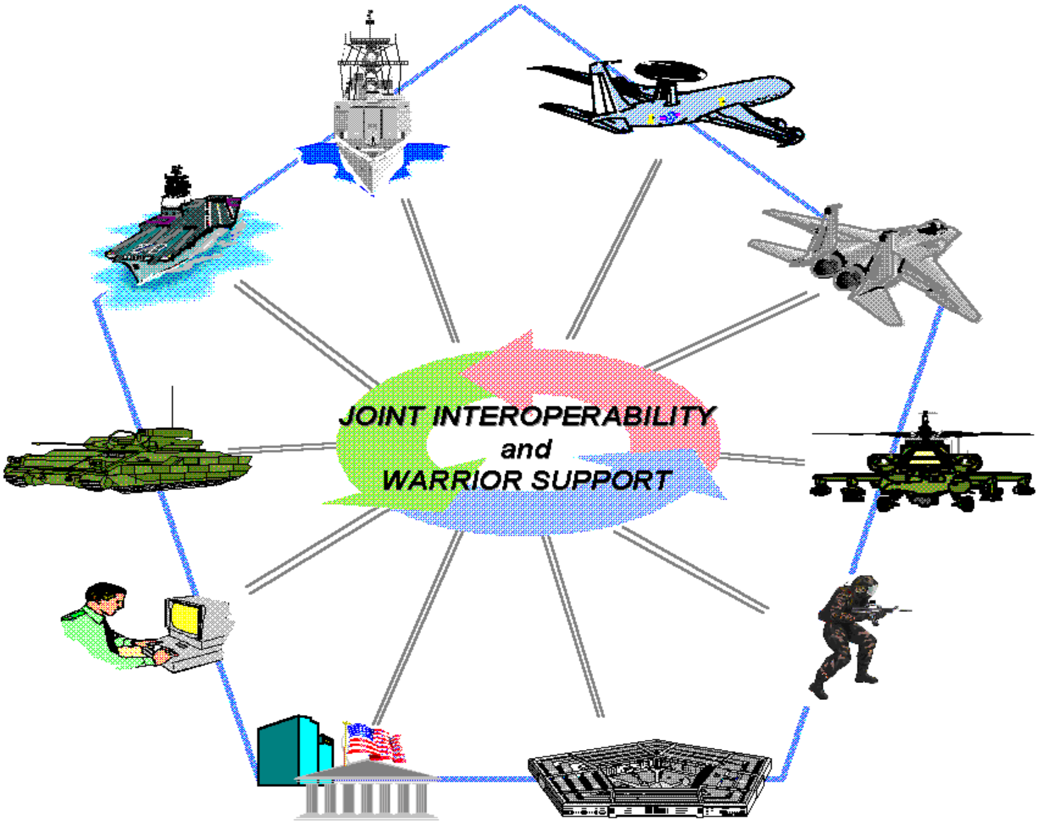
Department of Defense

Technical Reference Model



Version 1.0

5 November 1999

DISTRIBUTION STATEMENT A: Approved for public release; distribution unlimited

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

SECTION 1: INTRODUCTION

1.1 PURPOSE

The purpose of the Department of Defense (DoD) Technical Reference Model (TRM) described in this document is to provide a common conceptual framework, and define a common vocabulary so that the diverse components within DoD can better coordinate acquisition, development, interoperability, and support of DoD information systems. The DoD TRM document provides an extensive set of service definitions and relationships that can be used to increase interoperability and compatibility between systems, as well as resolve related issues. The Technical Reference Model has also been enhanced with an extensive set of interfaces to support the identification and resolution of issues where real-time and performance considerations are of importance. This document, together with the model's two enhanced views (i.e., services and interfaces), presents extensive support for interoperability across a broad range of DoD applications and requirements.

The DoD TRM provides guidance to developers, system architects, and individuals in using and developing systems and technical architectures. The model promotes open system design but is not a system architecture. The TRM establishes a common vocabulary and defines a set of services and interfaces common to DoD systems. The reference model provides the foundation for the organization and structure for technical architectures. The reference model and technical architecture support the operational architecture and become key drivers for the systems architecture.

The use of the DoD TRM can:

- Facilitate and enable interoperability
- Enable portability and scalability
- Support open systems concepts
- Promote product independence and software reuse
- Facilitate manageability

1.2 SCOPE

The DoD TRM can be applied to all DoD information systems and information technology applications at all DoD organization levels and environments (e.g., tactical, strategic, sustaining base, interfaces to weapons systems). The DoD TRM provides insight and guidance in the development of technical and system architectures that satisfy requirements across missions, and in particular where interoperability and open systems issues are encountered. The DoD TRM guides the selection of interfaces and services in support of a technical architecture. The model provides a basis for addressing interoperability issues relative to service and interface definitions.

1.3 APPLICABILITY

The DoD TRM is to be considered as the foundation model describing services, interfaces and their interrelationships that can be applied to all systems, including multi-platform, networked, and distributed applications. Services and Agencies are encouraged to apply the model to support interoperability, portability, open systems, promte reuse and reduce life-cycle cost.

1.4 HISTORY

On 16 November 1990, the Secretary of Defense directed the implementation of the DoD Corporate Information Management (CIM) initiative, known as the DoD Information Management initiative, to strengthen DoD's ability to effectively apply computing, telecommunications, and information-management capabilities to accomplish the DoD mission. Transitioning DoD's present information systems and associated information technology resources to a communications and computing infrastructure based on the principles of open-systems architecture and systems transparency is a key strategy for implementing the Department's Information Management initiative. Developing a technical reference model and selecting associated standards are first steps toward executing this strategy and developing a technical architecture. This DoD TRM is a composite evolutionary model derived from the Technical Architecture Framework for Information Management (TAFIM) TRM Volume 2 and the Society of Automotive Engineers (SAE) Generic Open Architecture (GOA) Model. It is a foundation piece of such initiatives as the Joint Technical Architecture (JTA), the Defense Information Infrastructure (DII), and other Defense and Federal programs.

The development of this document is driven in part by the JTA's request to respond to their need for a more effective technical reference model capable of supporting IT and real-time system requirements. The document is also driven by the need for a single DoD TRM that can be updated in a more timely manner than the traditional standards activities cycle time for developing new standards. By declaring this document as a guide and not a standard, its utilitarian value is increased and its organization insertion time is minimized.

Both DII and JTA documents draw directly from the service definitions and groupings contained in the TAFIM TRM, which also serve as the basis for services in the DoD TRM. To ensure consistency, the service structure of the TAFIM TRM was imported into the DoD TRM.

1.5 DOCUMENT ORGANIZATION

The DoD TRM document consists of five sections and seven appendices. Section 4 is the key section in this document, describing the DoD TRM. The remaining sections complement and embellish other aspects relating to the DoD TRM. Section 2 provides justification for the use of a Technical Reference Model. Section 3 provides a high-level overview of three historical reference models. Section 4 contains a description of the enhanced DoD TRM and contains all of the service and interface definitions. Section 5 provides guidance in using the model. References are identified in Appendix A. Appendix B contains related and historical memoranda concerning the TRM. Abbrevations/Acronyms and glossary terms are defined in Appendix C. Appendix D contains an alphabetic listing of services at two different levels of detail. This appendix is to be used as an aid in locating services, their contents, and definitions. Appendix E contains the relationship (i.e., mappings) to the DII Common Operating Environment and the JTA. Appendix F provides a DoD TRM case study. Appendix G includes a template for commenting on this document.

1.6 ACKNOWLEDGMENT

The TRM Working Group (TRMWG) is chaired by the Defense Information Systems Agency (DISA) Center for Information Technology Standards (CFITS) and was convened (February 1997) as part of a DISA initiative to improve and enhance the TAFIM TRM (Volume 2). Subsequently, the continual emergence of TRM issues across the Joint Technical Architecture (JTA) community resulted in the TRMWG also being designated as a TRM Joint Technical Architecture Development Group (JTADG) advisory group. In this capacity, the TRMWG was tasked to provide insight and written commentary to the JTADG in its role as a TRM consultant and body of expertise. Subsequently, a JTADG TRM Ad Hoc Focus Group was created to support the JTA process. The JTA Ad Hoc Focus Group works closely with the TRMWG.

The TRMWG comprises DoD members from the Services and Agencies, other Federal agencies, industry, and other TRM-knowledgeable sources. Many of the TRMWG members wear dual hats in that they are also members of the JTADG. Current TRMWG members represent the following Services and Agencies: Army; Navy; Air Force; Marine Corps; DISA; Office of the Secretary of Defense (OSD); Defense Modeling and Simulation Office (DMSO); Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR); Weapon Systems Technical Architecture Working Group (WSTAWG); Defense Airborne Reconnaissance Office (DARO); Ballistic Missile Defense Office (BMDO); National Imagery and Mapping Agency (NIMA); National Reconnaissance Office (NRO); Society of Automotive Engineers (SAE); academia and industry.

Essential to completion of the DoD TRM work were two key references worthy of mention: SAE GOA Standard, SAE AS4893, Generic Open Architecture Model; and NASA document CR 188269 Space Generic Open Avionics Architecture (SGOAA) model. These two reference documents provided valuable information on the interface views contained in the DoD TRM.

SECTION 2: TECHNICAL REFERENCE MODELS

2.1 BACKGROUND

When two or more systems or components are required to interoperate or exchange information, a set of common and consistent service and interface definitions is needed to ensure the integrity of the information to be passed or exchanged. The set of definitions, integrated into a framework or abstraction, is known as a reference model. The need for a reference model is now recognized by the Information Technology community. In retrospect, the importance of such a set of definitions was not fully realized by the broader DoD community while systems remained isolated or compartmented in domain specific functional areas. The realization became even more apparent when consistent means of adjudicating interoperability, architectural, open-systems, and standard selection issues were needed by system architects and developers. Rapid changes in technology and the need to provide extensive battlefield coordination and effect joint operations have further underscored the need for such a set of definitions associated around a model. In the absence of a common model, DoD Services and Agencies were left no choice but to develop their own domain reference models in an effort to satisfy their requirements and users. The net effect of model proliferation over time has been confusion, model inconsistencies, and barriers in the DoD Joint Staff effort to achieve greater and more effective interoperability. The need for a foundation model that provides greater definition and clarity of services and interfaces is essential if a DoD open-systems approach conducive to interoperability is to be achieved.

The intent of the Technical Reference Model is to minimize continued proliferation of domain models in support of open-systems and interoperability across domains, in joint operations, and across a wide range of applications.

2.2 TRM CONCEPT

Diverse and demanding DoD requirements have resulted in the emergence of a number of different technical reference models or model variants across the Services (e.g., Army, Navy) and Agencies (e.g., DISA, NIMA, NASA). These models contain many common elements between them, (e.g., service and interface definitions, similar structures, and a significant overlap in functional capabilities). Furthermore, the source of these different models or variants from an earlier common high-level abstraction (i.e., NIST and IEEE reference model), suggests that a common representation may be possible to counter model proliferation by the Services and Agencies. An examination of Service-domain functional models reveals that they share many common elements (i.e, entities, services, interfaces) that provide the basis to develop a common representation or higher-level abstraction model in which interoperability can be supported on a larger scale. It would be in DoD's best interests to develop a high-level interoperability model, such as the DoD TRM, that can satisfy different operational domains such as those identified by the Joint Technical Architecture (e.g., C4ISR, Weapon Systems, Modeling and Simulation, and Combat Support).

In the past, custom systems were developed for specific hardware platforms using proprietary systems software (e.g., operating system, text editor, file management utilities). Such customization was necessary because DoD requirements were unique relative to those found in the commercial marketplace. These systems were not designed to interoperate with other systems nor to be portable to other hardware platforms. In addition, different systems were developed to perform similar functions at different levels of the overall DoD organization (national, theater, and unit) and for the different Services (Army, Navy, Air Force, Marine Corps). As a result, many of the systems proved to be functionally redundant with those of other applications. This situation often hindered systems evolution toward greater interoperability, data sharing, portability, and software reuse.

Proper attention to, and application of, the DoD TRM will assist organizations in achieving more effective levels of portability and interoperability in the following ways:

- Interoperability requirements are described in a standard way.
- Consistent specification of system architecture.
- Support for commonality across systems.
- Consistent use of standards.
- Comprehensive identification of interfaces.

Any such model must also be evolutionary and flexible enough to support current as well as future needs across a broad range of requirements and platform configurations. The model must be tailorable to enable users to extract only those elements required to support their domain needs and to exploit technology. The set of services and interfaces must also be robust enough and malleable to enable system architects and developers to develop their domain-specific views. The DoD TRM is intended to fill that role.

2.2.1 Interoperability Requirements are Described in a Standard Way

Information exchange and interoperability requirements between DoD systems can be described in terms of the model's vocabulary and the particular layer of the model affected by the requirement. The use of the DoD TRM may influence the description of requirements in such a way that standards may emerge for describing interoperability requirements.

Using the DoD TRM, systems can be defined in terms of consistent definitions and common functionality (i.e., via the services and interfaces). The functionality is needed to provide interoperability, portability, and scalability of computerized applications across networks of heterogeneous hardware/software platforms. Information exchange and interoperability requirements between DoD systems can be described in terms of the model's vocabulary and structure. The model may even be used to develop guides for describing interoperability requirements.

2.2.2 Consistent Specification of System Architecture

The DoD TRM facilitates developing system architectures that specify the characteristics of key system interface requirements and ensure that these requirements and the system and technical "responses" are clearly related to each other across all views of the architecture (operational, system, and technical). The application layer, for example, should be defined primarily in support of application interoperability and portability.

The system architecture can be consistently specified to ensure that open system requirements and interoperability considerations are addressed. For example, the External Environment Interface (EEI) is the interface between the application platform and the external environment across which information is exchanged. It is defined primarily in support of system and application software interoperability. Data administration defines the standardization and registration of individual data element types to meet the requirements for data sharing and interoperability among information systems throughout the enterprise.

2.2.3 Support for Commonality Across Systems

The DoD TRM facilitates the development of a common infrastructure to support interoperability and portability of applications. The DoD TRM guides the implementation of a communications and computing infrastructure based on open systems and common interoperability and including, but not limited to, operating systems, database management, data interchange, network services, network management, and user interfaces. The basis for a common infrastructure is to identify core capabilities that are applicable across multiple services. This, in the near term, enables the migration from static and monolithic applications (stovepipes) to a more open environment enabling data and data format transparency across different platforms.

2.2.4 Consistent Use Of Standards

Use of the DoD TRM facilitates the grouping of standards, as in the Joint Technical Architecture (JTA).

2.2.5 Comprehensive Identification of Interfaces

The DoD TRM, via its combination of service and interface views facilitates the identification and definition of a comprehensive set of interface specifications. This is needed to support development of complex DoD systems.

2.3 JTA RELATIONSHIP

The DoD TRM is foundation of the JTA as well as of other initiatives such as the DII. The TRM is the key source of service and interface definitions that provide part of the JTA document structure and the accompanying service descriptions. Some parts of the JTA, (e.g., domain annexes), prefer to use and reference the DoD TRM interface views in specifying their requirements. The ability of the DoD TRM to support different types of system requirements and different views is illustrated by the varied use of the model within the DoD community and within the JTA document.

Both the JTA and the DoD TRM are examining the impact of enlarging their respective additional domains beyond information technology-oriented command-and-control and sustainment domains. The DoD TRM and JTA are being expanded to accommodate real-time embedded weapons and avionics system domains, as well as modeling and simulation domains. These domains traditionally have systems or components that require carefully engineered, certifiable, real-time performance requirements. In order to keep the DoD TRM current, a number of different views within the same model are required from which a number of more specific domain-oriented representations can be derived. These representations are also capable of supporting real-time system development concerns more effectively.

2.4 OPEN SYSTEMS/REUSE

One of the key benefits expected from consistent TRM application is a significant increase in development efficiency due to software and system component reuse. Part of the reason for this increase in the efficiency in developing DoD systems will be due to the use of standards-based open system environments. These standards-based environments will facilitate the reuse of software, hardware, and processes used to solve problems within a domain.

Establishing standards-based open system environments that accommodate the injection of new standards, technologies, and applications will occur on a DoD-wide basis. These standards-based environments provide the basis for development of common applications that can be reused to provide similar functionality in other systems.

Many of the systems developed within a domain (such as command and control systems or intelligence systems) provide many of the same functions in most of the implementations of the concept. Most command and control systems, for example, provide message processing, message composition, communications, and situation-monitoring functions (which often include geopositional display capabilities). Most of these functions are now being provided by off-the-shelf hardware and software, which is a mix of commercial products and Government-developed software.

For those applications that must be custom-developed, incorporating software reuse into the development methodology will reduce the amount of software developed and add to the inventory of software suitable for reuse by other systems. Developers will look for similar applications that can be easily modified to meet new requirements or, in limited cases, develop new software using some components from a reuse library. Rare is the case, in true reuse environments, where an application must be completely developed from scratch.

The DoD TRM provides a basis for consistency in viewing, discussing, and configuring these reusable components in a wide range of systems.

2.5 ARCHITECTURE

The DoD TRM is a useful structuring tool in the development of the three major DoD-defined architecture views defined by the C4ISR Architecture Framework: operational, system, and technical.

An Operational Architecture (OA) view describes the tasks and activities, operational elements, and information flows required to accomplish or support a military operation. The OA identifies the operational needs of the warfighter at the application level of abstraction. It defines the types of information exchanged, the frequency of the exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges in detail sufficient to ascertain specific interoperability.

A System Architecture (SA) view can be defined at various levels; however, in DoD it is primarily a high-level description of the systems and interconnections that support the warfighter function. The SA shows how multiple systems link and interoperate and may describe the internal construction and operations of particular systems within the architecture. For the individual system, the SA view includes the physical connection, location, and identification of key nodes (including materiel item nodes), circuits, networks, warfighting platforms, etc., and it specifies system and component performance parameters (e.g., mean time between failure, maintainability, availability). The SA view associates physical resources and their performance attributes to the operational view and its requirements following standards defined in the technical architecture.

A Technical Architecture (TA) view is an established set of enterprise wide system implementation guidelines upon which engineering specifications are based, common building blocks are built, and product lines are developed. The primary purpose of a TA is to define a collection of the technical standards, conventions, rules, and criteria that govern system implementation and system operation. A TA is intended to be the minimum set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements. It is based on associations between operational requirements and their supporting systems, enabling technologies and interoperability criteria.

Each of the above defined architectures is influenced by the DoD TRM, which provides a pivotal reference in defining key concepts for services and their interfaces. The TRM serves to provide DoD architects with a common reference point as they proceed through the various phases of defining their architecture models, and for structuring the development of detailed architectural products. Additional detail on the definitions and interdependencies of OA, SA, and TA views can be found in the C4ISR Architecture Framework, Version 2.0, dated 18 December 1997.

SECTION 3: HISTORICAL TECHNICAL REFERENCE MODELS

3.1 INTRODUCTION

This section presents the three key models—Portable Operating System Interface (POSIX), Technical Architecture Framework for Information Management (TAFIM) TRM, and Generic Open Architecture (GOA)—that serve as the foundation for developing and enhancing the Department of Defense (DoD) Technical Reference Model (TRM) described in Section 4. The DoD TRM adopts the structure of the POSIX reference model and combines it with the service view of the TAFIM TRM and the interface view of the GOA model into a single uniform and tailorable model. The objective of the model is to provide a complete and more extensive set of model features and capabilities. The DoD TRM thus provides consistency of user application by a broader defense community to address interoperability, open systems, acquisition, and management issues associated with commercial off-the-shelf (COTS) and Government off-the-shelf (GOTS) products.

3.2 POSIX GENERIC MODEL

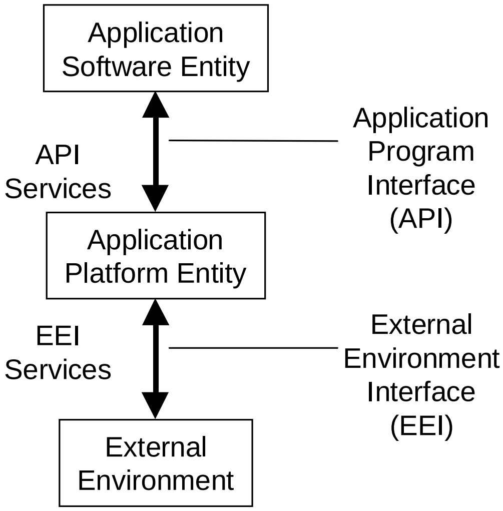
The generic DoD Technical Reference Model is a set of concepts, entities, interfaces, and diagrams that provides a basis for the selection of standards. To a large extent, the DoD TRM adopts the foundation work of the International Organization for Standardization (ISO) Open Systems Environment (OSE) Reference Model Working Group as reflected in their ISO/ International Electro-Technical Commission (IEC) Technical Report 14252 - 1995 (IEEE 1003.0

- 1995), Guide to the POSIX Open System Environment. Within the guide, an interface is defined as “a shared boundary between the two functional units.” The functional units are referred to as “entities” when discussing the classification of items related to application portability.

The basic elements of the generic DoD TRM are those identified in the POSIX OSE Reference Model and are presented in Figure 3.2-1. As shown in the figure, the model includes three classes of entities and two types of interfaces as follows:

- Application Software Entity.
- Application Program Interface (API).
- Application Platform Entity.
- External Environment Interface (EEI).
- External Environment.

This model has been generalized to the point that it can accommodate a wide variety of general- and special-purpose systems. More detailed information is presented in subsequent sections; however, the service specifications allow for subsets or extensions as needed.


Application Software Entity
API Services
Application Program Interface (API)
Application Platform Entity
EEI Services
External Environment Interface (EEI)
External Environment

Reference: IEEE Guide to the POSIX Open System Environment
(IEEE 1003.0-1995), Institute of Electrical and Electronics Engineers, Inc.,1995

Figure 3.2-1 POSIX Open System Environment Reference Model

From the perspective of the application software entity, these services are provided by an application platform whether the particular services are provided from the local platform or from remote platforms that may comprise one or more nodes of a larger distributed system. The generic model can also be applied in a distributed environment.

3.3 TAFIM TECHNICAL REFERENCE MODEL (TRM)

3.3.1 TAFIM TRM Overview

The TAFIM TRM, Version 3.0, in support of DoD's Information Technology initiatives, provides a common conceptual framework and defines a common vocabulary to assist the diverse components within DoD. TAFIM TRM Version 3.0 represents the latest and most current version approved by DoD for dissemination across the CINCs, Services, and Agencies, and is defined as, and represented in, TAFIM Volume 2. With the exception of the two sets of interfaces defined—application program interfaces and external environment interfaces—the model represents a service-centric view (i.e., focusing on the services and their definitions).

3.3.2 TAFIM Description

Figure 3.3.2-1 expands upon Figure 3.2-1 to present the DoD TRM entities and interfaces, including the service areas of the Application Platform and related services. Figure 3.3.2-1 depicts only entities, interfaces, and service areas and does not imply interrelationships among the service areas. Later versions decomposed this top entity into two sub-entities: a Mission-Area Applications sub-entity and a Support Applications sub-entity. This was done in part to distinguish between unique mission-support software applications and common applications that can be standardized across individual or multiple mission areas. This decomposition convention has been carried into the DoD TRM.

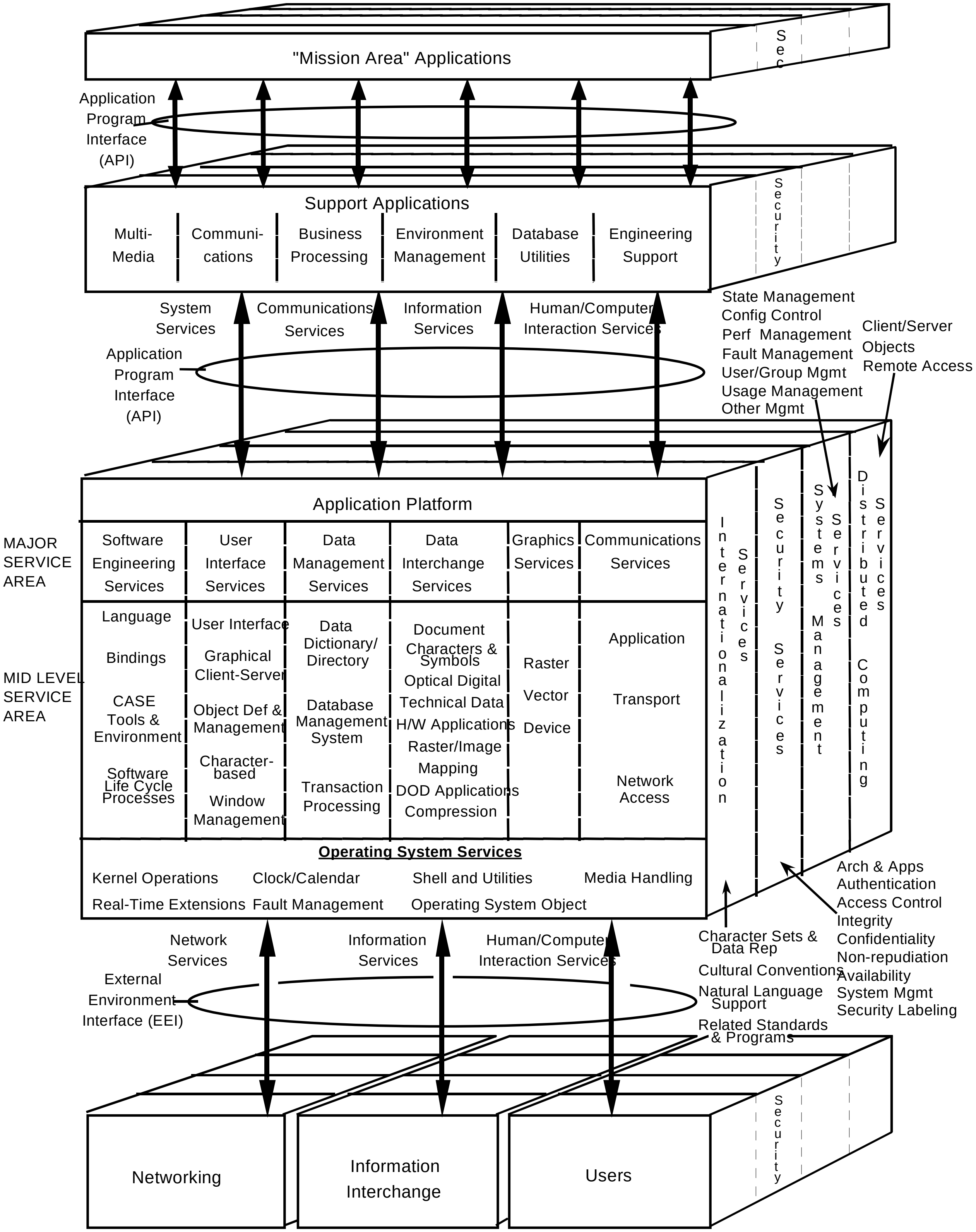


Figure 3.3.2-1 Detailed TAFIM Technical Reference Model

Users should assess their own requirements and create a profile of services, interfaces, and standards that satisfy their own mission-area needs. Users who have adopted earlier versions of the figure should consider adopting the new version of the figure only when planning a major revision of their documentation

3.3.2.1 Application Software Layer

The model decomposes the Application Software Entity into two sub-entities: a Mission-Area Applications sub-entity that implements specific end-user requirements or needs; and a Support Applications sub-entity that represents common applications that can be standardized across individual or multiple mission areas. Support Applications and their services can be made available to the user or can be used to develop mission-area-specific applications

3.3.2.2 Application Platform Service Areas

This section provides a characterization of the terms used to describe the Application Platform Service Areas of the TRM. These terms provide a common definition for the services and interfaces used by DoD information systems and apply to all volumes of the TAFIM. The TAFIM describes the information technology (IT) services provided by the Application Platform Service Area in three levels of detail: Major Service Area (MSA), Mid-Level Service Area (MLSA), and Base Service Area (BSA).

Each major heading (MSA) establishes a grouping of services or functionality defined by industry standards and is expressed in a way to be consistent with the manner in which the standards bodies are addressing these groups. The sub-headings (MLSA and BSA) identify more specific, concrete examples of the functionality represented by the major grouping.

This distinction of major mid- and base-level service areas, has not been carried through in the DoD TRM. The focus of the DoD TRM is the concentration and availability of a common set of definitions and relationships that can be used to define interoperability and open-system needs.

3.4 GENERIC OPEN ARCHITECTURE (GOA) FRAMEWORK

3.4.1 GOA Introduction

The Generic Open Architecture (GOA) Framework is a reference model developed by the Avionics Systems Division (ASD) of the Society of Automotive Engineers (SAE). The SAE GOA model evolved from the NASA Space Generic Open Avionics Architecture (SGOAA) model, as documented in NASA CR 188269. The SAE GOA model is documented as a standard in publication SAE AS4893 *"Generic Open Architecture (GOA) Framework."* Additional information associated with the GOA Framework standard is documented in publication SAE AIR5315 "*Rationale and Overview of Generic Open Architecture (GOA) Framework Standard."*

3.4.2 GOA Overview

The GOA Framework defines a processing node reference model within a system of interconnected and inter-related processing nodes. Figure 3.4.2-1 graphically depicts the Generic Open Architecture (GOA) Framework as illustrated in the SAE GOA standard. This reference model defines a generic set of four hierarchical layers of functionality with each layer composed of components termed entities. The GOA Framework further defines nine classes of interface points mapped to the generic model layers. The instance of a complete set of layers with associated interface points on a given processing node is termed a GOA Stack. The composite of the lower three layers is comparable to the applications platform as defined by the IEEE POSIX reference model and TAFIM TRM.

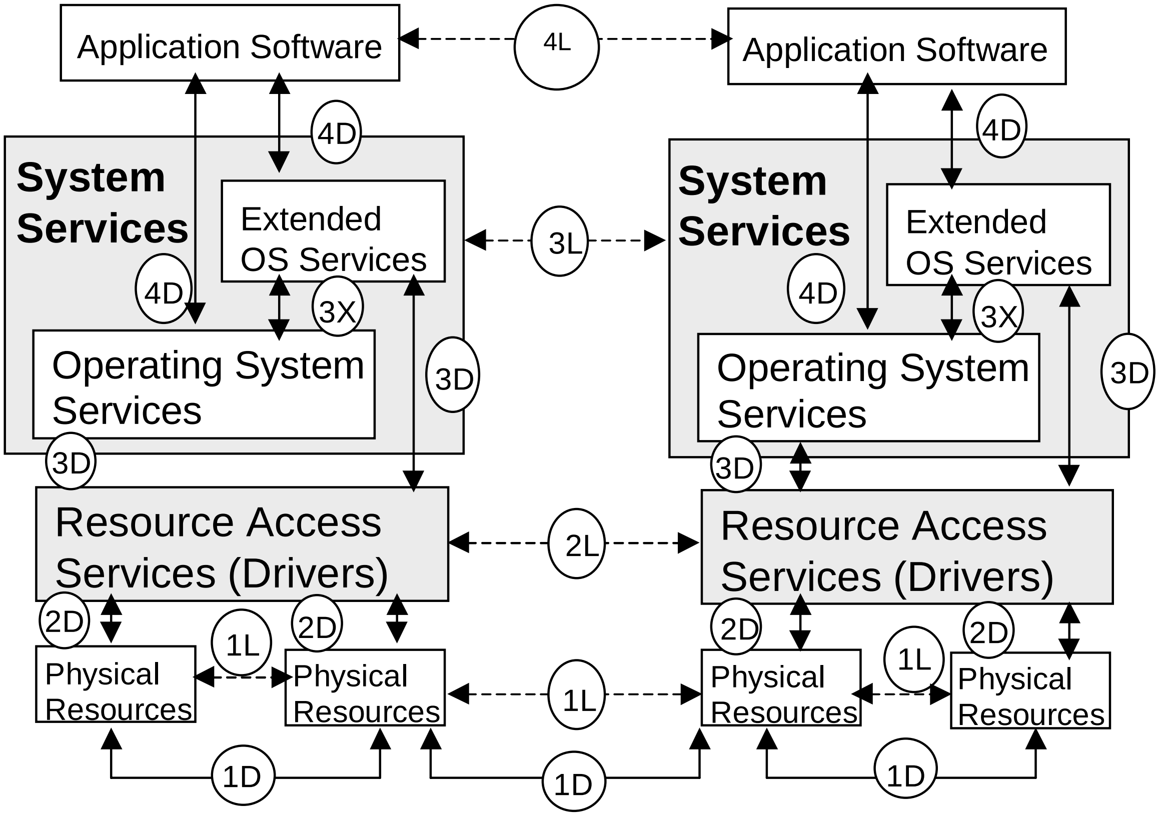


Figure 3.4.2-1 Placement of Interface Classes within the GOA Framework (Two Platform View)

The GOA Framework defines 4 hierarchical layers and is distinctive in its definitions of direct and logical interfaces.

3.4.2.1 GOA Layers

The GOA Framework layers are briefly defined as follows:

> **Layer 4** is the highest hierarchical layer and is termed the Applications Software layer. This layer corresponds to the DoD TRM Application Software entity described in paragraph 4.4.1 and Figure 4.5-1.
>
> **Layer 3** is the top software isolation layer directly below the Software Applications layer and is termed the System Services layer.
>
> **Layer 2** is also an isolation layer and is termed the Resource Access Services layer. It is composed of low level services or functional entities, typically referred to as "device or hardware function drivers," which abstract hardware entity details from layer 3.
>
> **Layer 1** is the lowest hierarchical layer and is termed the Physical Resources layer. It may be composed of just hardware entities, but could also include firmware and very low level software.

Layer 1, Layer 2, and Layer 3 form the Application Platform or "host" for Layer 4 Applications software, in the DoD TRM, Figure 4.5-1.

These layers are interconnected via direct (e.g., 4D, 3X, 3D, 2D, and 1D) and logical (e.g., 4L, 3L, 2L, and 1L) interfaces that define their relationships.

3.4.2.2 GOA Interface Classes

Logical Interface: A GOA Framework interface, dashed lines in Figure 3.4.2-1, is the specification among peer entities that supports the understanding necessary for sharing of information (objects, data, parameters, status, and control) among those entities, and can be independent of the exchange transport mechanisms and media between them.

Direct Interface: A GOA Framework interface, solid lines in Figure 3.4.2-1, is the specification among hierarchical entities that supports the physical transfer of information (objects, data, parameters, status, and control) among those entities.

These logical and direct interfaces are fully defined in the SAE AS4893 standard document.

3.5 SUMMARY AND CONCLUSIONS

3.5.1 General Correspondence Between the TAFIM TRM and GOA Framework

The GOA is a hierarchical framework compatible with the TAFIM TRM, both consisting of: Application Software, System Services sub-layered into Operating System services and Extended Operating System Services, Resource Access Services, and Physical Resources. Figure 3.5.1-1 shows the high-level correspondence of the TAFIM TRM layers to the GOA Framework.

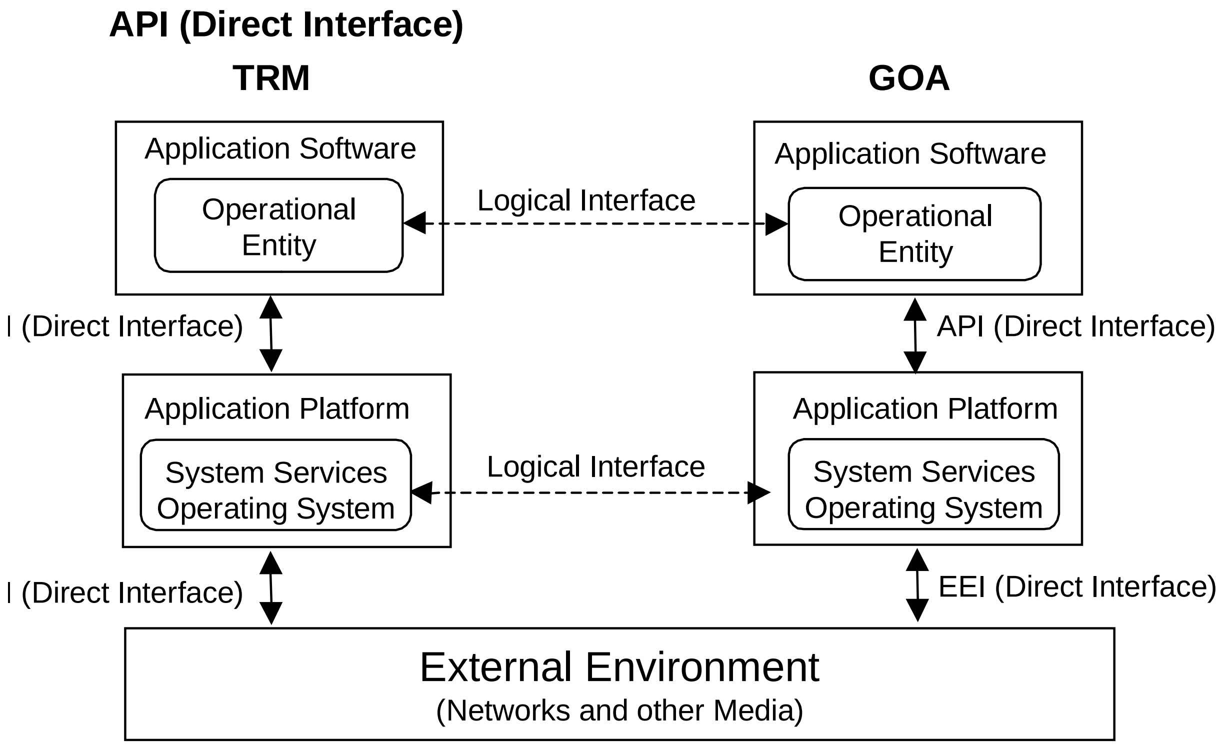


Figure 3.5.1-1 TRM and GOA Layer Correspondence

In Figure 3.5.1-1, the TAFIM TRM layering has been enriched with the notion of logical and direct interfaces as a precursor to considering GOA interface classes. The notion expressed in Figure 3.5.1-1 is that operational entities communicate via logical interfaces. The semantics of those communications are the result of invoking some set of underlying services whose service entities communicate through direct interfaces. Logical and Direct interfaces are the coarsest distinction in the GOA. Direct interfaces are those that define a service/consumer relationship between adjacent layers in the GOA model. Direct interfaces are those involved with sending, routing, and receiving data between two or more entities. Logical interfaces define peer-to-peer relationships between entities within the same GOA layer. Logical interfaces are the GOA-requirement associated with establishing a data interchange between a source of data and its destination. Therefore, if the GOA layers are mapped to the TRM layers and the TRM data flow is known, a GOA-compliant instance of the TAFIM TRM should be derivable. Figure 3.5.1-1 reveals an important principle of the GOA class organization: Logical Interfaces to Services are implemented via Direct Interfaces. In order to map the partitions of logical and direct interface to the layers of the architecture, the relationships between application software, system services, and resource services (drivers) will need to be elaborated.

3.5.2 Conclusion

Services and their accompanying definitions are used to ensure consistency and equivalency between two or more system components required to communicate (i.e., establish an accurate alignment of information sources to destinations). The extensive nature of GOA interfaces, together with the elaborate set of TAFIM service definitions, provides a viable set of parameters from which interoperability and open system needs can be defined and established to effectively support operational and mission needs. The DoD TRM described in Section 4 represents a combination of the best features and elements of the TAFIM TRM and the SAE GOA model.

The DoD TRM represents an update to, and hence replacement of, the TAFIM TRM as the TAFIM is rescinded from policy. The DoD TRM model uses the interface elaboration and conventions of the SAE GOA to define those interfaces contained within its APIs and EEIs. Adoption of the SAE GOA convention facilitates mapping of the interface views with the service views across models. In this manner, TAFIM TRM users, SAE GOA users, and DoD TRM users can easily map services and interfaces from their respective model or system to either legacy or newly emerging models to ensure interoperability.

SECTION 4: DOD TECHNICAL REFERENCE MODEL.

4.1 NEW MODEL CONCEPT

The Department of Defense (DoD) Technical Reference Model (TRM) integrates the service view and interface view of previous models to meet the requirements of increasingly diverse and complex systems. The DoD TRM can be tailored to support a wide range of requirements, based in part on the following characteristics of the model:

- Ability of the model to support system architectures so that migration, enhancement, and technology insertion efforts can be supported.
- Degree(s) of freedom enabled by the model to select and or expand on services and interfaces.
- Ability of the model to support and allow new service and interface definitions, associations, and environment configurations (i.e., network, distributed, platform-centric, multiplatform, and decentralized).
- Ability of the model to present or support different views (e.g., services only, interfaces only, services and interfaces, functional).
- Ease of mapping the model to other known reference models to facilitate establishment of relationships and links.

4.2 STRUCTURE AND MULTIPLE VIEWS

The model presented in Figure 4.2-1 is based on the same IEEE POSIX three-entity model discussed earlier in Section 3.2. For the sake of clarity and discussion, the three-entity model is duplicated and shown as the six major entity blocks contained in Figure 4.2-1 with corresponding interfaces identified between entity blocks. This basic variation can be transformed to the various configurations contained in all of the other models found in DoD. This model is a non-intrusive model in that other lower-level models, currently being used across DoD, are not impacted and can be easily derived from this model. This single model thus establishes a means of communications (linking) between models to resolve differences in their structure. The Figure 4.2-1 model enables the identification of direct and logical interfaces between the entities (discussed in greater detail in Section 4.5). Representations of distributed, networked, client-server, and multi-processor configurations can also be clearly depicted using the diagram and adding additional entity blocks as needed. These representations were always implied in the documented DoD models (e.g., TAFIM TRM), but not clearly depicted.

Figure 4.2-1 enables the association and assignment of respective services and service groups within each of the major block (entity) designations. Elaborating on the services and major service groups and their mapping into the six major entities is intentionally deferred at this time. This is in recognition of the fact that some models, at the next level of decomposition, will further decompose or break-out entities differently depending upon their domain of application or requirements. For example, in the TAFIM TRM the top most entity—the Application Software Entity—is broken down into a Mission-Area Application and a Support Application. This is primarily done to separate or identify mission-specific end-user requirements or needs. Thus, it is premature to assign services to an entity-level until the final entity level decomposition occurs.

The high-level view presented in the six-block model enables assignment of services and interoperability discussions to occur in a neutral setting, in the absence of polarizing views that occur when services are assigned to major service groups or sub-entity levels. An effective reference model, as a minimum, should assist in establishing common definitions, relationships,

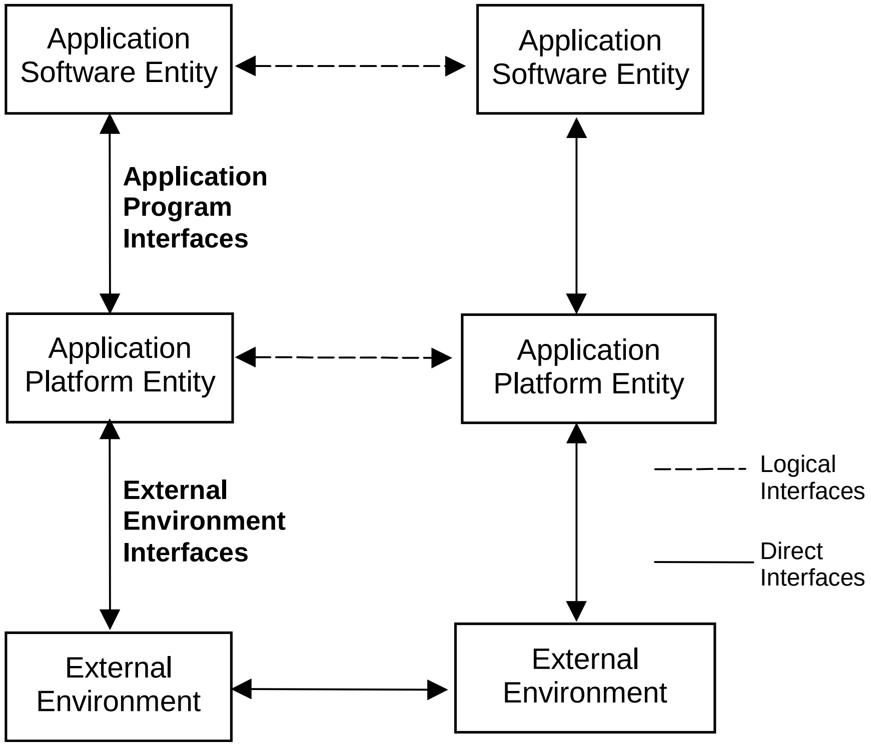

Application Software Entity
Application Software Entity
Application Program Interfaces
Application Platform Entity
Application Platform Entity
External Environment Interfaces
Logical Interfaces
Direct Interfaces
External Environment
External Environment

Figure 4.2-1 High Level Representation of the DoD TRM

and associations for purposes of understanding interoperability issues and subsequently the selection of standards and reusable components. The high-level representation model can also be used to group respective services and interfaces of other existing models to establish domain boundaries and equivalency. The latter supports transformations between models. This high-level model is intended to enable communications from one DoD community and their model to another. The model is not a “new” replacement model nor is it an architecture, but it should be viewed as an aid to assist in addressing interoperability issues and as a refinement to the TAFIM TRM.

Key features of the high-level representation DoD model shown in Figure 4.2-1 are the model’s abilities to:

- Provide for individual tailoring to existing and emerging user requirements and needs.
- Enable transformations from one model or view to another for purposes of understanding semantics and intended use.
- Enable the identification and inclusion of new services and interfaces that may be needed due to performance, technology, and hardware demands.
- Provide a common and independent DoD framework to address interoperability issues of C4ISR and Weapons Systems.

- Associate service and interface groups across more than one platform (since the figure can represent more than one application platform).
- Examine and address interoperability issue (since the model can be applied to commercial models external to DoD).

The DoD TRM represents an enhanced and harmonized model incorporating the best characteristics of several models to support a broad range of DoD systems, both real-time and non-real-time.

4.3 MAJOR ELEMENTS

The three major elements of the model are (1) the services, (2) the interfaces and (3) the entities that contain the services and interfaces. Entities have interfaces to other entities and the specific interface nature is determined by the configuration established by the user. Apart from core services and interfaces, the contents and organization of an entity are left up to the user and tailoring requirements within the limits of the entity definition.

4.3.1 Entities

Entities are the elements of the model that contain the services and interfaces subsequently used to select and refine a set of standards. Entities contain the major service areas which may be common across several entities. The DoD TRM does not distinguish below the major service area since these service associations are unique to specific models. Lower-level services and interfaces form the basis for defining and tailoring multiple model views.

4.3.2 Services

The service is a key common element that establishes a link (i.e., common denominator) between requirements, standards, and the supporting environment (tools). Services support requirements via their functions and capabilities, while standards implement services. The service definitions contained in this document represent an extensive set derived from key reference model documents (see Appendix A) and coordinated via the Technical Reference Model Working Group (TRMWG) and the Services. The service descriptions are intended to provide consistent definitions that can be applied across the different domains of application. They are intended to provide uniform means of establishing agreement and consensus between the different users on what a service is, and what is and is not included in its set of supporting functions and capabilities. The existence of an extensive set of core services does not mandate that a particular model or user include all or use all of those services identified in the core set. Users may select the set of core services applicable to them.

It is also recognized that some services may be included in more than one major service area due to user requirements or technology drivers that necessitate such. Services may encompass both hardware and software capabilities. As the list of services is expanded, additional guidance on their use will be included in this document. However, the existence of a core service with its accompanying definition should not be modified indiscriminately. Users must utilize that definition or justify why it should be modified before they are free to invent new service definitions for an existing one.

4.3.3 Interfaces

To support a broad range of system capabilities and performance requirements (inclusive of real-time needs) an extensive set of interfaces is required within the model. Interfaces represent the “pipes” through which all services are provided. The interface provides the “connective tissue” between the entities in Figure 4.2-1. These “pipes” are represented by a set of horizontal and vertical interfaces that fall into two major categories: logical and direct. Logical interfaces define what information is exchanged, i.e., they represent establishment of a data interchange between a source of data and its destination. Logical interfaces define peer-to-peer relationships between similar entities (at the same level of abstraction). Direct interfaces define relationships between adjacent entities and are those “directly” involved with the transmission, receipt, and routing of data between the entities. Direct interfaces allow the identification of operating system-to-extended-operating-system interfaces essential in weapon systems. The identification and type of interface is determined by user requirements and the operational domain in which a system is to be deployed.

The Figure 4.2-1 configuration thus allows for a robust and complete description of services and interfaces needed to effectively and adequately describe (tailor) a user’s needs across a broad range of systems, applications, or platform configurations. The figure also identifies various configurations that a system may require in supporting its operational mission in both the classified and unclassified mode.

4.3.4 DoD TRM Infrastructure

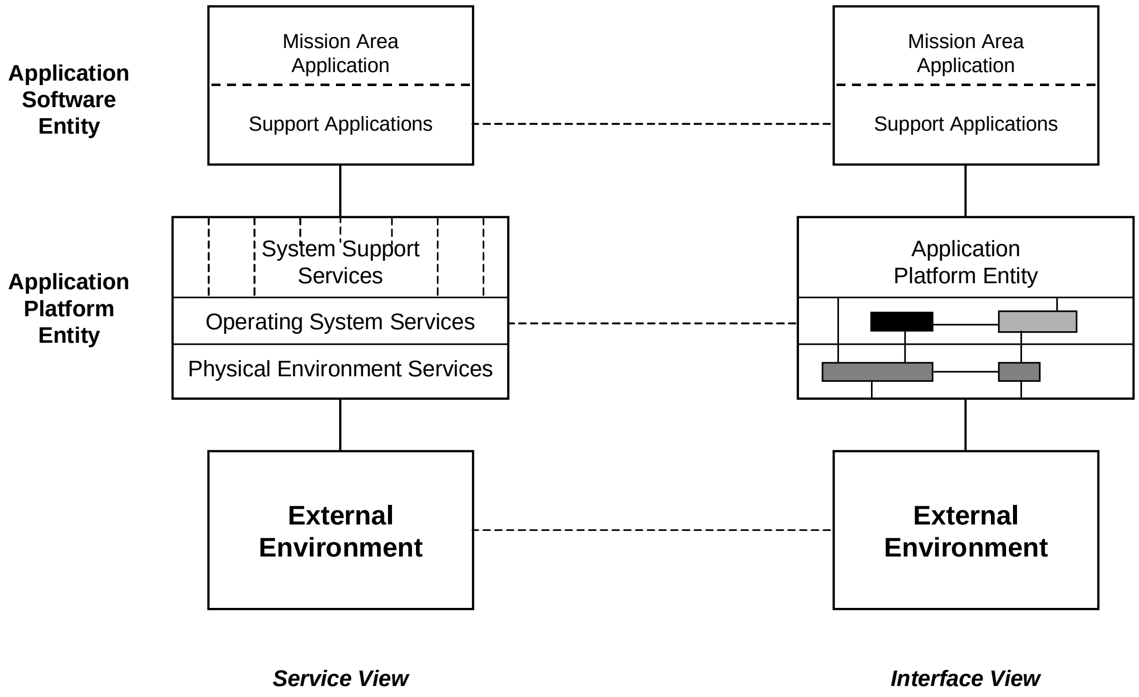


Figure 4.3.4-1 DoD TRM Multi-platform View Showing both Services and Interfaces

Figure 4.3.4-1 is the same basic model shown in Figure 4.2-1 with additional entity decomposition to illustrate that either a service or interface view, or both, can be graphically represented at the user's discretion. The figure also shows the tailorable nature of the model. This enhanced model enables coexistence of a service view with an interface view. This depiction can provide accurate service relationships and present specific logical and direct interfaces where required for any domain. Figure 4.3.4-2 shows the DoD TRM services view of Figure 4.3.4-1, identifying examples of service areas. The representation also allows for future addition of other services or interfaces should they be required without changing the intent or semantics of the model.

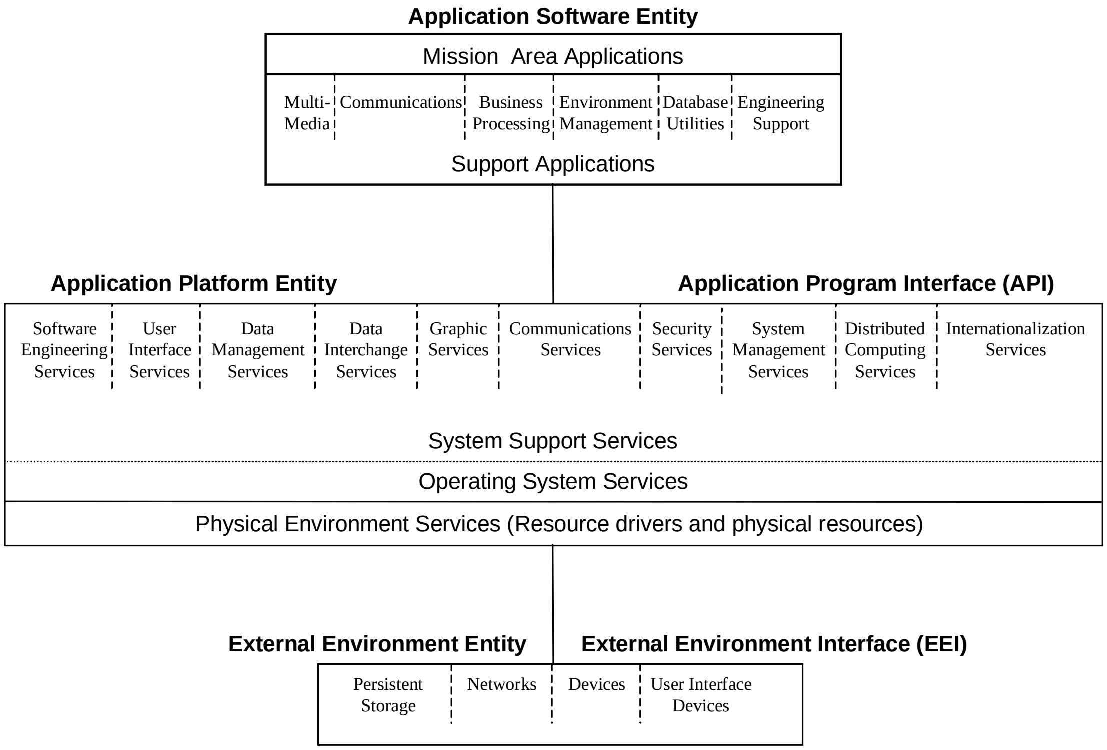

Application Software Entity
Mission Area Applications
Multi-Media
Communications
Business Processing
Environment Management
Database Utilities
Engineering Support
Support Applications
Application Platform Entity
Application Program Interface (API)
Software Engineering Services
User Interface Services
Data Management Services
Data Interchange Services
Graphic Services
Communications Services
Security Services
System Management Services
Distributed Computing Services
Internationalization Services
System Support Services
Operating System Services
Physical Environment Services (Resource drivers and physical resources)
External Environment Entity
External Environment Interface (EEI)
Persistent Storage
Networks
Devices
User Interface Devices

Figure 4.3.4-2 DoD TRM Service View

4.4 SERVICE AREAS

The initial delineation of major service areas, such as those identified in Figure 4.3.4-2, and their orientation (i.e., horizontal or vertical) is presented to assist readers in associating a particular service with an entity. Services, or a subset of the services may migrate from one entity level to another depending on the application, and its commonality and use within an enterprise domain. Similarly, a service or its subset may exist in more than one model entity depending on the application or view. For example, security service components may be distributed or co-exist across several entities depending on how the security is implemented in terms of responsibility, levels of trust and compartmentalization, and operating-system architecture.

Figure 4.3.4-2 identifies major service areas commonly found within specific entities. The major service areas and the associated sub-services in any model are established as a product of the tailoring process in which requirements and specific user needs are identified.

Some models at higher levels of abstraction (e.g., major service area) label services differently (e.g., Communication Hardware Services versus Physical Services); but their lower sub-service levels consist of exactly the same components. For this reason, establishing equivalency between different models or views must be done at the sub-service (lower-level detailed service definition) level. Figure 4.3.4-2 is used to begin the user's tailoring process in support of a tailored view. It provides a framework that enables communications between users and identifies major service and interface boundaries and constraints.

The DoD TRM consists of an initial set of services (i.e., high-level or major service areas), and contains or points to lower-level services (i.e., domain specific). Lower-level services form the basis for defining and tailoring multiple views. A clear distinction must be made between those services and elements that are to be included, and those that will be excluded or not used when a view is selected.

4.4.1 Application Software Entity Services

The DoD TRM promotes the goals for developing modular applications and promoting software reuse to support the broad range of activities integral to any organization. To satisfy these goals, functional (mission-area) applications development will, in many respects, become an integration activity as much as a development activity. Application development will likely be accomplished by dividing and/or consolidating common functional requirements into discrete modules. Previously developed reusable code or Government off-the-shelf (GOTS) applications that could satisfy some, if not all, of the new functional requirements would be identified. Such reusable code/applications would then be integrated, to the extent possible, to become the software pieces necessary to complete the mission and/or support applications that will satisfy all of the requirements. In the DoD TRM, applications software can be divided into mission-area applications and support applications.

4.4.1.1 Mission-Area Applications

Mission-area applications implement specific end-user requirements or needs (e.g., sensor, payroll, accounting, materiel management, personnel, control of real-time systems, analysis of order of battle). This application software may be commercial off-the-shelf (COTS) or GOTS, custom-developed, or a combination of these. In addition to application software, an information system may include data that can be application-specific (e.g., a log of invoices and payments) or an integral software part (e.g., application parameters, screen definitions, diagnostic messages).

4.4.1.2 Support Applications

The set of services described in this section provides initial capabilities that will be used to define, acquire, and develop common, shared applications. The services have been grouped into service areas categorized by function. The service areas and services will most likely change over time. New services will be added or, in some cases, existing services will be rearranged and merged into new service areas. Some of the services, particularly those found in the multimedia category, will be used as building blocks to implement other services. An implementation of a support application may actually merge several services from several different areas.

Support applications are common applications (e.g., e-mail, word processing, spreadsheets) that can be standardized across individual or multiple-mission areas. The services they provide can be used to develop mission-area-specific applications or can be made available to the user. Support applications may be COTS products selected to provide a service in a common manner, or they may be GOTS applications developed to meet a DoD-unique need and reused in multiple systems. The Defense Information Infrastructure (DII) Common Operating Environment (COE) includes several support applications to provide common functions such as message handling, network browsing, and mapping. For example, the Joint Mapping Toolkit (JMTK) provides objects and services to support geospatial analysis, mapping (visual) display, geospatial database management, and image preprocessing.

4.4.1.2.1 Multimedia Services

Multimedia services provide the capability to manipulate and manage information consisting of text, graphics, images, video, and audio. These services can be used directly by mission-area applications, but they can also be used by other support applications to satisfy a common requirement. These services can be used in combination or separately. Multimedia services include:

Text-processing services, including the capability to create, edit, merge, and format text.

Document-processing services, including the capability to create, edit, merge, and format documents. These services enable the composition of documents that incorporate graphics, images, and even voice annotation, along with stylized text. Included are advanced formatting and editing services such as style guides, spell-checking, use of multiple columns, table-of-contents generation, headers and footers, outlining tools, and support for scanning images into bit-mapped formats.

Electronic-publishing services, including incorporation of photographic-quality images and color graphics, and advanced formatting and style features such as wrapping text around graphic objects or pictures and kerning (i.e., changing the spacing between text characters). These services also interface with sophisticated printing and production equipment.

Geographic information system (GIS) services, including the capability to create, combine, manipulate, analyze, and present geospatial information. This includes the creation of entity symbology that overlays the map background display and access to standard symbol libraries.

Image-processing services providing for the capture, scan, creation, and edit of images in accordance with recognized image-formatting standards.

Video-processing services, including the capability to capture, compose, and edit video information. Still graphics and title-generation services are also provided.

Audio-processing services, including the capability to capture, compose, and edit audio information.

Multimedia-processing services, including the capability to compress, store, retrieve, modify, sort, search, and print all or any combination of the above-mentioned media, and to perform these actions on two or more types of media simultaneously. This includes support for microform media, optical-storage technology that allows for storage of scanned or computer-produced documents using digital storage techniques, a scanning capability, and data compression. Additionally, multimedia processing includes hypermedia processing. Hypermedia provides the capability to create and browse documents that allow users to interactively navigate through the document using information embedded in the document.

4.4.1.2.2 Communications Services

Communications services provide the capability to send, receive, forward, and manage electronic and voice messages. They also provide real-time information exchange services in support of interpersonal conferences. These services include:

Personal-messaging services, including the capability to send, receive, forward, store, display, and manage personal messages. This includes the capability to append files and documents to messages. Messages may include any combination of data, text, audio, graphics, and images and should be capable of being formatted into standard data-interchange formats. This service includes the use of directories and distribution lists for routing information, the ability to assign priorities, the use of pre-formatted electronic forms, and the capability to trace the status of messages. Associated services include a summarized listing of incoming messages, a log of messages received and read, the ability to file or print messages, and the ability to reply to or forward messages.

Organizational-messaging services, including the capability to send, receive, forward, display, retrieve, prioritize, and manage predefined and unformatted organizational messages. Organizational messages should use standard data-interchange formats and may include any combination of data, text, audio, graphics, and images. This includes the capability to review and authenticate messages. Incoming message-processing services include receipt, validation, distribution, and dissemination of incoming unformatted messages based on message profiling, message precedence, and system-security restrictions. User support services include the selection and display of messages from a message queue, online management of search profiles, search and retrieval of stored messages based on message content comparison to queries formulated by the analysts, and composition of record messages for transmission. Outgoing message-processing services include coordination by the command’s staff organizations, authorized release, and verification of record messages prior to transmission.

Enhanced telephony services, including call-forwarding, call-waiting, programmed directories, teleconferencing, automatic call distribution (useful for busy customer service areas), call detail recording, and voice mail.

Shared-screen teleconferencing services allow two or more users to communicate and collaborate using audioconferencing with common “shared” workstation windows that refresh whenever someone displays new material or changes an existing display. Every user is provided with the capability to graphically annotate or modify the shared conference window.

Videoconferencing services that provide two-way video transmission between different sites. These services include full motion display of events and participants in a bi-directional manner and support for the management of directing the cameras, ranging from fixed-position, to sender-directed, to receiver-directed, to automated sound pickup.

Broadcast services that provide one-way audio or audio/video communications services between a sending location and multiple receiving locations.

Computer conferencing services that allow groups to participate in conferences via computer workstations. These conferences may not occur in real-time. Conferees or invited guests can drop in or out of conferences or sub conferences at will. The ability to trace the exchanges is provided. Services include exchange of documents, conference management, recording facilities, and search-and-retrieval capabilities.

4.4.1.2.3 Business Processing

Business support services provide common office functions used in day-to-day operations. Business support services include:

Spreadsheet services, including the capability to create, manipulate, and present information in tables or charts. This capability should include fourth-generation-language-like capabilities that enable the use of programming logic within spreadsheets.

Project management services, including tools that support the planning, administration, and management of projects.

Calculation services, including the capability to perform routine and complex arithmetic calculations.

Calendar services, including the capability to manage personal tasks and time and to coordinate multiple personal schedules via an automated calendar.

4.4.1.2.4 Environment Management

This type of service is broader in scope than the other categories in that it exists primarily to manage a particular data-processing and/or communications environment. Environment management services integrate and manage the execution of platform services for particular applications and users. These services are invoked via an easy-to-use, high-level interface that enables users and applications to invoke platform services without having to know the details of the technical environment. The environment management service determines which platform service is used to satisfy the request and manages access to it through the API.

Batch-processing services support the capability to queue work (jobs) and manage the sequencing of processing based on job-control Commands and lists of data. These services also include support for the management of the output of batch processing, which frequently includes updated files or databases and information products such as printed reports or electronic documents. Batch processing is performed asynchronously from the user requesting the job.

Transaction-processing services provide support for the online capture and processing of information in an interactive exchange with the user. This typically involves predetermined sequences of data entry, validation, display, and update or inquiry against a file or database. It also includes services to prioritize and track transactions. Transaction-processing services may include support for distribution of transactions to a combination of local and remote processors.

Information presentation and distribution services are used to manage the distribution and presentation of information from batch and interactive applications. These services are used to shield mission-area applications from how information is used. They allow mission-area applications to create generic pools of information without embedding controls that dictate the use of that information. Information presentation and distribution services include the selection of the appropriate formatting services required to accomplish the distribution and presentation of information to a variety of mission-area applications, support applications, and users. It also includes the capability to store, archive, prioritize, restrict, and recreate information.

Learning Technology (LT) services include computer-based training (CBT), computer-assisted instruction (CAI), intelligent tutoring, and distance learning. Learning technology services provide for an enterprisewide integrated training environment. CBT and CAI services support a type of education in which the student learns by executing special computer-based training programming. This includes tutorial training on the application in use and the availability of offline, onsite interactive training. Distance Learning services support a type of education in which students work on their own and communicate with faculty and other students via e-mail, electronic forums, videoconferencing, and other forms of computer-based communication.

4.4.1.2.5 Database Utilities

Database utility services provide the capability to retrieve, organize, and manipulate data extracted from a database management system. These common services provide a consistent interface to the user while providing access to a variety of databases. Database utility services include:

Query-processing services that provide for interactive selection, extraction, and formatting of stored information from files and databases. Query-processing services are invoked via user-oriented languages and tools (often referred to as fourth-generation languages), which simplify the definition of searching criteria and aid in creating effective presentation of the retrieved information (including use of graphics). Fourth-generation languages are generally all proprietary. Some are in the public domain (for example, Dbase clones are generally referred to as “Xbase” systems), but these all started as proprietary systems. As yet, no public domain fourth-generation language is in wide business use.

Screen-generation services that provide the capability to define and generate screens that support the retrieval, presentation, and update of data.

Report-generation services that provide the capability to define and generate hardcopy reports composed of data extracted from a database.

Networking/concurrent access services that manage concurrent user access to database management system (DBMS) services.

4.4.1.2.6 Engineering Support

Engineering support services include support for analysis, design, modeling, development, and simulation for a wide variety of users and environments. This includes computer-aided design (CAD) services for designing, drafting, and producing engineering drawings. It also includes services provided by decision support development tools and expert system shells.

CAD services provide high-precision drawing tools and modeling capabilities to allow production of engineering specification drawings and other precise drawings.

Decision support services provide interactive modeling and simulation tools that support analysis of alternative decisions.

Expert system services provide artificial intelligence capabilities usually based on knowledge- or rules-based inference engines that recommend or take actions based on presented situations and prior “experiences.”

Modeling and simulation services provide the capability to capture or set object characteristics or attributes and parameters of a system of objects, and to portray the relationships and interactions of the objects to assist in the analysis of the system.

4.4.2 Application Platform Entity Services

The application platform is defined as the set of resources that support the services on which application software will execute. It provides services at its interfaces that, as much as possible, make the implementation-specific characteristics of the platform transparent to the application software.

To ensure system integrity and consistency, application software entities competing for application platform resources must access all resources via service requests across the API. Examples of application platform services may include an operating system kernel, a real-time monitor program, and all hardware and peripheral drivers.

The application platform concept does not imply or constrain any specific implementation beyond the basic requirement to supply services at the interfaces. For example, the platform might be a single processor shared by a group of applications, a multi-processor at a single node, or it might be a large distributed system with each application dedicated to a single processor.

The application platform implementations that use the TRM may differ greatly depending upon the requirements of the system and its intended use. It is expected that application platforms defined to be consistent with the TRM will not necessarily provide all the features discussed here, but will use tailored subsets for a particular set of application software.

4.4.2.1 System Services

The following services are system services. Some services have dual roles, such as security, internationalization, system management services, and distributed-computing services.

Extended operating system services are needed to support mission needs not readily available from the existing commercially available operating systems or are services built on low-level operating system interfaces that could be privileged. This could require that a kernel service or operating system service be extended and may be needed by the Mission Application or Applications.

4.4.2.1.1 Software Engineering Services

Professional system developers require tools appropriate to the development and maintenance of applications. These capabilities are provided by software engineering services, which include:

Language services that provide the basic syntax and semantic definition for use by a software developer to describe the desired application software function. Shell and executive script language services enable the use of operating-system commands or utilities rather than a programming language. Shells and executive scripts are typically interpreted rather than compiled, but some operating systems support compilers for executive scripts. Other programming tools may use procedural or object-oriented languages to define the functionality of the desired applications. Third-generation languages provide primarily command-line interfaces and text-based code for defining the applications, while more recent fourth-generation languages are forms-based and provide a graphical interface.

Bindings and object code linking provide the ability for programs to access the underlying application and operating system platform through APIs that have been defined independently of the computer language. They are used by programmers to gain access to these services using methods consistent with the operating system and Specific language used. Only Ada refers to such actions as “language bindings.” All other compilers, DBMSs, and system software refer to such actions as “linking.” Linking is operating-system-dependent, but language-independent.

Computer-Aided Software Engineering (CASE) tools and environment include systems and programs that assist in the automated development and maintenance of software. These include, but are not limited to, tools for requirements specification and analysis, for design work and analysis, for creating and testing program code, for documenting, for prototyping, and for group communication. The interfaces among these tools include services for storing and retrieving information about systems and exchanging this information among the various components of the system development environment. An adjunct to these capabilities is the ability to manage and control the configuration of software components, test data, and libraries. Other fourth-generation language tools include software development tools such as artificial-intelligence tools and the UNIX command “imake.”

Software life-cycle processes identify distinct phases in the software life-cycle, which is the period of time that begins when a software product is conceptualized and ends when the software is no longer available for use. It includes a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals). The software life-cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and the retirement phase.

4.4.2.1.2 User Interface Services

User interface services define how users may interact with an application. They provide a consistent way for people who develop, administer, and use a system to gain access to applications programs, operating systems, and various system utilities. The user interface is a combination of menus, screen design, keyboard commands, command language, and help screens, which create the way a user interacts with a computer. User interface services make use of drivers for mice, touch screens, and other input hardware. Depending on the capabilities required by users and the applications, these interfaces may include the following:

Graphical client-server operations define the relationships between client and server processes operating within a network, in particular, graphical user interface display processes. In this case, the program that controls each display unit is a server process, while independent user programs are client processes that request display services from the server.

Object definition and management services, which define characteristics of display elements such as color, shape, size, movement, graphics context, user preferences, and interactions among display elements.

Character-based user interface, which can be either a command-line interface or a menu-driven interface similar to a graphical user interface, but it does not use graphics and may depend solely on the keyboard for user input, i.e., not make use of an explicit pointing device. Modern systems and applications are and will continue to be based upon graphical user interfaces and the associated standards for such systems. However, many legacy systems still include a large number of character-based terminals and interfaces.

Window management specifications, which define how windows are created, moved, stored, retrieved, removed, and are related to each other.

Within the past few years, significant advances have been made in user interfaces, both in ease of use and in reducing the development effort required. Common frameworks, aids to the document and defined functional goals, and requirements for application designers can be found in such references as Appendix A, item #21, TAFIM Vol. 8, Human-Computer Interface; and item # 22, DoD Joint Technical Architecture. Although other technologies can be used, most users think of a user interface in terms of a graphical user interface (GUI). A GUI allows a user to specify actions by dragging and dropping or pointing and clicking on an icon that is a pictorial metaphor for the object being acted upon. A GUI can also depict several actions simultaneously by presenting multiple windows.

The services associated with a windows system include the visual display of information on a screen that contains one or more windows or panels, support for pointing to an object on the screen using a pointing device such as a mouse or touch-screen, and the manipulation of a set of objects on the screen through the pointing device or through keyboard entry.

4.4.2.1.3 Data Management Services

Central to most systems is the management of data that can be defined independently of the processes that create or use it, maintained indefinitely, and shared among many processes. Data management services include:

Data-dictionary/directory services, which allow data administrators and information engineers to access and modify data about data (i.e., metadata). Such data may include internal and external formats, integrity and security rules, and location within a distributed system. Data-dictionary/ directory services also allow end users/applications to define and obtain data that are available in the database. Data administration defines the standardization and registration of individual data element types to meet the requirements for data sharing and interoperability among information systems throughout the enterprise. Data administration functions include procedures, guidelines, and methods for effective data planning, analysis, standards, modeling, configuration management, storage, retrieval, protection, validation, and documentation.

Database management system services, which provide data administration, managed objects functionality, and controlled access to, and modification of, structured data. To manage the data, the DBMS provides concurrency control and facilities to combine data from different schemas. Facilities may also include the capability to manage data in a distributed computing environment where data are stored on multiple, heterogeneous platforms. DBMS services are accessible through a programming language interface, an interactive data manipulation language interface such as SQL, or an interactive/fourth-generation language interface. For efficiency, database management systems generally provide specific services to create, populate, move, backup, restore/recover, and archive databases, although some of these services could be provided by general file management capabilities described in operating-system services.

- **Transaction-processing** services, which support the definition and processing of “transactions.” A transaction is a “unit of work” consisting of a series of operations that must be completed together.

4.4.2.1.4 Data Interchange Services

Data interchange services provide specialized support for the interchange of information between applications and to/from the external environment. These services are designed to handle data interchange between applications on the same platform and applications on different (heterogeneous) platforms.

Document interchange services are supported by specifications for encoding the data (e.g., text, pictures, numerics, special characters) and both the logical and visual structures of electronic documents. Services support document exchange between heterogeneous computer systems, exchange of military-formatted messages, and electronic forms interchange.

Characters and symbols services provide for interchange of character sets and fonts and standardized date and time representation.

Optical digital technologies (ODTs) represent technologies that use the reflective properties of light and an optical-recording surface to capture, encode, decode, and store data. ODT predominantly encompasses optical media, optical drives, and scanners.

Technical data interchange services provide facilities for the exchange of technical data. This includes standards for the interchange of graphics data, typically vector graphics, technical specifications, and product data. Product data encompasses technical drawings, documentation, and other data required for product design and manufacturing, including geometric and non-geometric data such as form features, tolerances, material properties, and surfaces.

Hardware applications services provide data interchange services between non-homogeneous hardware components. The most common example of this service is the interchange of information between a computer and a printing device. These services include font information exchange, bar coding, optical-disk handling, and graphics device interface (GDI) APIs.

Raster/image data interchange services provide for the handling and manipulation of raster graphics and images. Raster graphics standards are standards for pixel-by-pixel representation of images. Image data standards are standards for the exchange of imagery data, metadata, and attachments to the images.

Mapping services provide formats and facilities for machine-readable mapping, charting, and display of geospatial data.

Compression services specify algorithms for compressing data for storage and exchange over a network. Data compression can reduce communications loading by as much as 80 percent without affecting the form of transmitted data. Compression requires application of the same algorithms at the sending and receiving locations. Compression may be used for text and data, still images, and motion images. Compression algorithms for data must be “lossless” so that the expanded output exactly matches the original input. Compression algorithms for still and motion images may be “lossy,” where some data may be lost, but the expanded output is not noticeably different from the original input.

4.4.2.1.5 Graphics Services

Graphics services provide functions required for creating and manipulating pictures. These services include:

Raster graphics, which represent images as a matrix of dots. Raster graphics images are created by scanners and cameras and are generated by paint software packages. The simplest monochrome bitmap uses one bit (on/off) for each dot. Gray-scale bitmaps (monochrome shades) represent each dot with a number large enough to hold all the gray levels. Color bitmaps require sufficient storage to hold the intensity of red, green, and blue, as would a gray-scale equivalent.

Vector graphics, which represent graphical objects as sets of endpoints for lines, curves, and other geometric shapes with data about width, color, and spaces bounded by lines and curves. The entire image commonly is stored in the computer as a list of vectors called a display list. Vector graphics are used when geometric knowledge about the depicted object is needed. Geometric shapes keep their integrity—a line always can be separately selected, extended, or erased. Today, most screens are raster graphics displays (composed of dots), and the vectors are put into the required dot patterns (rasters) by hardware or software. Vector graphics systems must be supplemented by data interchange standards, such as Initial Graphics Exchange Specification (IGES), Computer Graphics Metafile (CGM), and the Standard for the Exchange of Product Model Data (STEP).

Device interfaces provide API services for accessing graphics devices, such as monitors, scanners, printers, etc.

4.4.2.1.6 Communications Services

Communications services are provided to support distributed applications requiring data access and applications interoperability in networked environments.

Application services are the functions and interfaces that reside on the underlying network and communications system protocol software and are used by applications. These services are based on the presentation and application layers (layers 6 and 7) of the OSI Reference Model.

Transport services perform a variety of functions concerned primarily with the end-to-end transmission of data across a network and end-to-end reliability. The services performed include end-to-end error detection and recovery, regulating flow control, and managing the quality of service. Transport services correspond to the transport and session layers (layers 4 and 5) of the OSI Reference Model.

Subnetwork technologies services support access to local area networks (LANs) and other networks based on the physical, data link, and network layers (layers 1, 2, and 3) of the OSI Reference Model. This area includes LANs, point-to-point communications, packet-switching, circuit-switching, and military-unique data communications.

4.4.2.1.7 Security Services

Multilevel security cuts across all aspects of the system and adds an additional complexity to the hardware and software that interacts with the rest of the system. This could be a special feature of the hardware and software and can be multi dimension such as Application level, kernel level, device level, system level, and application platform level.

Different groups of individuals within and across the various DoD mission areas need to work with specific sets of data elements. Access to these sets of data elements is to be restricted to authorized users. Satisfaction of this requirement generally has been accomplished by the implementation of separate information systems. Organizations cannot continue to afford to implement separate information systems to satisfy this requirement, nor is it effective to require the user to change interface components every time the need arises to operate with a different restricted data set. Significant benefit will be realized when an individual information system can effectively support the needs of different groups of users and data sets. Such an information system will allow multiple groups to share information systems and data while guaranteeing the separation of data and users as necessary through the use of multi-level secure operating systems.

In multi-level secure operating systems, the kernel will play the prime role in permitting platforms to handle multiple information domains (security contexts) simultaneously. The separation kernel will be trusted software; that means it will be evaluated in accordance with the requirements stipulated in the documents cited in JTA Reference 31. The separation kernel will mediate all use of the basic information system resources and will provide for strict separation among multiple security contexts by creating separate address spaces for each of them. The separation kernel will provide separation among process spaces by using the protection features of the platform hardware (e.g., processor state registers, memory mapping registers).

Emerging security initiatives anticipate the expanding use of Non-Developmental Item (NDI) products in information system implementations. This view is reflected in such references in Appendix A, as item #21, TAFIM Volume 6, Defense Goal Security Architecture (DGSA); item #22, DoD Joint Technical Architecture, Version 2.0; and item #25, Defense Information Technology Security Certification Accreditation Procedure (DITSCAP). For this reason, two categories of software are identified: trusted and untrusted. Both categories may have been acquired for an information-system implementation as NDI products. However, the trusted software will have been evaluated in accordance with criteria established by responsible agencies for information-system security and will need to be maintained under strict configuration management control. Trusted software will mediate the access of all untrusted software to information-system resources. Such control, which the DGSA suggests should be in the operating system kernel, will provide the necessary security protection by maintaining separation among applications at different security levels that are simultaneously processing.

Security services are necessary to protect sensitive information in the information system. The appropriate level of protection is determined based upon the classification to the mission-area end users and the perception of threats to it. The information system integrator will need to work with the designated accreditation authority (DAA) to identify the required level of security protection and acceptable mechanisms for satisfying the requirements. Information system security services are depicted as cross-area services in Figure 3.3.2-1 because the mechanisms implemented to provide them may be part of multiple-platform service areas. The DGSA currently identifies implementations of security service protection mechanisms in the platform as part of the network and operating-system service areas.

The DGSA identifies the following security services that may need to be provided through implementations in information system components. The first five of these services are consistent with the definitions contained in ISO 7498-2, a standard focusing on security related to open systems interconnection communications. The DGSA extends the ISO 7498-2 definitions to apply to more than communications and identifies availability as a security service.

Authentication service ensures that system entities (processes, systems, and personnel) are uniquely identified and authenticated. The granularity of identification must be sufficient to determine the processes, system, and personnel's access rights. The authentication process must provide an acceptable level of assurance of the professed identity of the entities.

Access control services prevent the unauthorized use of information system resources. This service also prevents the use of a resource in an unauthorized way. This service may be applied to various aspects of access to a resource (e.g., access to communications to the resource; the reading, writing, or deletion of an information/data resource; the execution of a processing resource) or to all accesses to a resource. Security labels are used to manage access and privileges, which are managed for all entities, whether they are individual users, groups of users, resources, or processes.

Integrity service ensures protection of the system through open system integrity, network integrity, and data integrity. This ensures that data are not altered or destroyed in an unauthorized manner. This service applies to data in permanent data stores and to data in communications messages.

Confidentiality service ensures that data are not made available or disclosed to unauthorized individuals or computer processes through the use of data encryption, security association, and key management. This service will be applied to devices that permit human interaction with the information system. In addition, this service will ensure that observing usage patterns of communications resources will not be possible.

Non-repudiation services include open systems non-repudiation, electronic signature, and electronic hashing. Non-repudiation services ensure that senders and recipients cannot deny the origin or delivery of data. Non-repudiation mechanisms can be used to validate the source of software packages or to verify that hardware is unchanged from its manufactured state.

Availability service ensures that timely and regular communications services are available. These services are intended to minimize delay or non-delivery of data passed on communications networks. These services include protecting communications networks from accidental or intentional damage and ensuring graceful degradation in communications service.

System management services encompass those security functions required to maintain an operationally secure system. These services include analysis areas such as certification and accreditation and risk management, as well as operationally motivated concerns such as alarm-reporting, audit, and cryptographic key management.

Security labeling is the data bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

Information security management services involve the installation, maintenance, and enforcement of information-domain and information security policy rules in the information system intended to provide these security services. In addition to these core services, security management requires event-handling, auditing, and recovery. Standardization of security management functions, data structures, and protocols will enable interoperation of security management application programs (SMAPs) across many platforms in support of distributed security management.

Classes of managed objects for security management are security policies, security services, and security mechanisms. Some information is managed for specific information domains and for the platform in a distributed or non-distributed environment. The items of information that might be included in the security management information base (SMIB) for each information domain and for the platform itself are described in TAFIM Volume 6.

4.4.2.1.8 System Management Services

Information systems are composed of a wide variety of diverse resources that must be managed effectively to achieve the goals of an open system environment. While the individual resources (such as printers, software, users, processors) may differ widely, the abstraction of these resources as managed objects allows for treatment in a uniform manner. The basic concepts of management, including operation, administration, and maintenance, may then be applied to the full suite of OSE components along with their attendant services.

Work on system management services and attendant standards is ongoing. This work is based predominantly on the Open System Interconnection (OSI) network management framework, which applies mainly to networks and the individual nodes on the networks. There is, however, an overlap among certain types of network management functions and individual system management functions. This overlapping area applies equally to networks and individual systems and forms the basis for the OSI approach to system and network management. Other system management functions in the typical operating system sense are also being addressed and need to be integrated into the overall systems and network management framework. System management functionality may be divided according to the management elements that generically apply to all functional resources: state management, configuration control, performance management, fault management, user/group management, usage management, and other management.

This breakout of system management services parallels the breakout of OSI network management, thereby presenting an overall coherent framework that applies equally to networks and the individual nodes of the networks. Many of the specific services have no formal standards work in progress; however, industry consortia and others are addressing selected areas.

One important consideration of the standards supporting the services in this area is that they should not enforce specific management policies but rather enable a wide variety of different management policies to be implemented, selected according to the particular needs of the end-user installations.

State management services provide for mechanisms that monitor, maintain, and change the state of the system or components of the system.

Configuration control services address four basic functions: identification, control, status accounting, and verification. Identification involves identifying and specifying all component resources. Control implies the ability to freeze configuration items and then to change them only through a process involving agreement of appropriate name authorities. Status accounting involves the recording and reporting of all current and historical data about each configuration item. Verification consists of a series of reviews and audits to ensure conformity between the actual configuration item and the information recorded about it. The services that provide these functions include software distribution and license management.

Performance management services allow information technology resources to be managed efficiently. Performance aspects of hardware, software, and network components must be monitored and subsequently made available to the system manager. The manager must then have access to services and parameters with which to tune the system to meet performance targets. This is accomplished through batch scheduling, system resource management, print and storage device management, system startup and shutdown, subsystem management, and communication of management information.

Fault management services allow a system to react to the loss or incorrect operation of system components at various levels (hardware, software, etc.). Fault management involves event management and network error recovery.

User/group management services provide traditional system administration interfaces for administering users and groups. These services are mechanisms for system and network administrators to use when implementing a management policy across a system. Administrators can use the services to establish domains and policies for management throughout the system. They can provide the ability for applications to access group and user databases. Users can set up their own areas of management and policies or use system defaults that are included in management services.

Usage management and cost allocation services include the management of software licensing, system cost management, and system resource allocation. Software license management for a system provides license administration, management, and enforcement services that allow more detailed, firm, and equitable licensing terms for users, and better protection against illegal software usage for vendors. Cost allocation services provide the ability to cost services for charging and reimbursement and to measure and prioritize resource usage. System resource allocation allows system administrators to control the amount of system resources available to users.

Other management services include the following services, which do not fit cleanly into any other management area: database administration, object-oriented database management, floppy disk formatting and handling, POSIX tape labeling and tape volume processing, and print management. Database and object-oriented database administration provide facilities and interfaces to manage databases and object-oriented databases, respectively. Floppy disk-formatting and -handling standards provide formats and interfaces for the exchange, backup, and restoration of data to or from floppy disks. POSIX tape labeling and tape volume processing provide standardized methods of handling and reading data stored on tape media and containing certain types of administrative information automatically readable by tape-handling software. Print-management services are used by management and user applications to send a file to a printer, cancel a print job, and get printer status information. (Security system management services are discussed in Section 4.4.2.1.7, as part of Security Services.)

System management application processes, using information in the information base, will be used to establish the required security contexts for interactive communications among distributed platforms operating in various information domains simultaneously. This approach is intended to support secure distributed-computing services. System management application processes will also be used to provide the security protection of store-and-forward communications in which the requisite security contexts cannot be handled within the message.

4.4.2.1.9 Distributed-Computing Services

Distributed-computing services provide specialized support for applications that may be physically or logically dispersed among computer systems in a network yet wish to maintain a cooperative processing environment. The classical definition of a computer becomes blurred as the processes that contribute to information processing become distributed across a facility or a network. As with other cross-cutting services, the requisite components of distributed-computing services typically exist within particular service areas. They are described below to offer a coherent view of this important service.

Client/server Services provide support for computing services partitioned into requesting processes (clients) and providing processes (servers), whether on the same platform or in a distributed environment.

Object Services support the definition, instantiation, and interaction of objects in a distributed environment, and include services that handle operating system bindings, message transport and delivery, and data persistence.

Remote-Access Services provide location transparency functionality for distributed-computing services, allowing users and client processes to access appropriate systems resources (files, data, processes) without regard to the location of either.

4.4.2.1.10 Internationalization Services

As a practice, information-system developers have generally designed and developed systems to satisfy a focused set of requirements relevant to a specific market segment. That specific market segment may be a nation or a particular cultural market. To make that information system viable, or marketable, to a different segment of the market, a full re-engineering process was usually required. Users or organizations that needed to operate in a multinational or multicultural environment typically did so with multiple, generally incompatible information-processing systems. NATO is an example in which a number of countries come together to work toward a common goal yet must deal with a diversity of languages and cultures in their day-to-day operations.

Within the context of the TRM, internationalization provides a set of services and interfaces that allow a user to define, select, and change between different culturally related application environments supported by the particular implementation.

Character sets and data representation services include the capability to input, store, manipulate, retrieve, communicate, and present data independently of the coding scheme used. This includes the capability to maintain and access a central character-set repository of all coded character sets and special graphical symbology used throughout the platform, including the appropriate modifications of GUI screens to match character-set conventions. Character sets will be uniquely identified so that the end user or application can select the coded character set to be used. This system-independent representation supports the transfer (or sharing) of the values and syntax, but not the semantics, of data records between communicating systems. The specifications are independent of the internal record and field representations of the communicating systems. Also included is the capability to recognize the coded character set of data entities and subsequently to input, communicate, and present that data.

Cultural convention services provide the capability to store and access rules and conventions for cultural entities maintained in a cultural convention repository. These repositories should be available to all applications and be capable of being sorted based upon local rules defined in the repository.

Native-language support services provide the capability to support more than one language simultaneously. Messages, menus, forms, and online documentation would be displayed in the language selected by the user. Input from keyboards that have been modified locally to support the local character sets would be correctly interpreted.

4.4.2.2 Operating System Services

Operating system services are the core services needed to operate and administer the application platform and provide an interface between the application software and the platform. Application programmers will use operating system services to access operating system functions. To separate sensitive data within an information system, the kernel must include mechanisms to control access to that information and to the underlying hardware. Operating system services include the following:

4.4.2.2.1 Kernel Operations

Kernel operations provide low-level services necessary to create and manage processes, execute programs, define and communicate signals, define and process system clock operations, manage files and directories, and control input/output processing to and from peripheral devices. Thread services provide an underlying service used for multiple concurrent executions within a single computer process. They are designed to allow independent operation and are essential for functions such as multiple-process communications.

4.4.2.2.2 Real-Time Extension Services

Real-time extension services support event-driven processes supporting management and actuation of physical processes. For this reason, they are often referred to as sensor-based systems. These services are designed to handle and process interrupts from a variety of sources (typically involving some kind of sensor device or timer), process associated information through some type of capture or control algorithm, and respond, if necessary, with an appropriate signal to a control or actuation device.

4.4.2.2.3 Real-Time Thread Extension Services

Supporting time-bound threads of control operations, this real-time behavior exploits inherent parallelism in underlying hardware, supports scheduling and the notion of an Ada task, transaction-processing, and networked/distributed systems. Threads are useful for mapping asynchronous behavior into equivalent synchronous behavior such as providing I/O parallelism controlling asynchronous computations, or structuring applications composed of many logically distinct tasks such as simulations and windowing systems.

4.4.2.2.4 Clock/Calendar Services

Clock/calendar services provide mechanisms for measuring the passage of time and maintaining the system time. This includes clocks and timers, real-time timers, and distributed timing services.

4.4.2.2.5 Fault Management Services

Fault management services include the prevention, isolation, notification, diagnosis, and correction of fault conditions, which arise whenever a malfunction or abnormal behavior results or may result in an error, outage, or degradation of services. Fault management services allow a system to react to the loss or incorrect operation of system components, and they encompass services for fault detection, isolation, diagnosis, recovery, and avoidance.

4.4.2.2.6 Shell and Utilities

Shell and utilities include mechanisms for services at the operator level, such as comparing, printing, and displaying file contents; editing files; searching patterns; evaluating expressions; logging messages; moving files between directories; sorting data; executing command scripts; scheduling signal execution processes; and accessing environment information.

4.4.2.2.7 Operating System Object Services

Operating system object services define the rules for creating, deleting, and managing objects.

4.4.2.2.8 Media-Handling Services

Media-handling services provide for disk and tape formatting for data and interchange of data with applications.

4.4.2.3 Physical Environment Services

Physical environment services are hardware-based services that include the interfacing software services provided by device drivers that support digital/analog signals between components. Some hardware devices have software (device drivers) embedded in them to enable the computer system with digital/analog timing sequences to operate in a correct manner. It should be noted that hardware-based services appear in two places in the DoD TRM: as part of physical environment services and as part of the external environment. The DoD TRM does not allocate any specific services to either one of these. The general notion is that services provided by the hardware that is part of a particular system are considered physical environment services, while the external environment provides the services of hardware outside that system. However, it is recognized that what constitutes a system is a matter of perspective. For example, in the context of a command-control system, the keyboard is sometimes considered to be part of the computer system, even if it is not inside the computer box. However, in a weapon system, which may make use of several computers in various functional subsystems, a keyboard at an operator station may be part of the weapon system, but it is external to any one of the specific computing subsystems. Even in the information systems context, components viewed by the end user as part of the system are frequently viewed by vendors of computers as external to it, even including components that may be physically housed inside the computer box. To some people, it may make a difference if a disk drive is internal or external, or even if the interface to it is part of the computer’s motherboard or on a separate plug-in board. Others will not make such distinctions.

For this reason, the following physical services contained in and supporting application platform services may also be found to exist in the external environment.

4.4.2.3.1 Devices

This highlights the general physical services required in an application platform. This includes, but is not limited to, the hardware interconnect services (e.g., backplanes), data storage services (e.g., tape and disk format standards), power supplies, temperature control, mechanisms and processing resources required to implement an application “platform.” Since one intent of open system standards is to provide independence of systems from the details of processing resources (i.e., details of CPU instruction sets), standards in this last area will generally not be specified. However, standards in the other areas are frequently required to support interoperability, portability, and technology upgrade goals of systems.

4.4.2.3.1.1 Backplanes and Buses

Backplanes and buses are hardware interconnect services that facilitate data transfer between physically separated systems, subsystems, and modules. At this level, information is frequently represented as changes in either voltage levels, current flow, or other physical parameters. However, existing standards also frequently consider logical interactions among components.

4.4.2.3.1.2 Storage

Hardware data storage services facilitate data retention. At this level, all information is represented as changes in either voltage levels, current flow, or other physical parameters. This area includes standards for the representation of data in physical storage media (e.g., disks, tapes, optical devices).

4.4.2.3.2 Hardware Processing

Hardware processing is the hardware service that manages, controls, and manipulates data.

4.4.2.4 Application Platform Cross-Area Services

Cross-area services are not distinguished as such in the DoD-TRM from the other services encountered within an entity. The extension or application of a service across more than one entity level is determined by user needs, the tailoring process, and requirements. In addition to the service areas delineated for the Application Platform (i.e., Section 4.4.2) there are other services that may affect the basic information system architectures within DoD. Treated in a manner similar to those in POSIX.0, these services may be referred to as cross-area or cross-entity services, and may have an effect on the operation of one or more services. Security, for example, may have an impact on more than one service within an entity level relative to the manner in which it is implemented or defined. In addition, there may be security safeguards whose requirements transcend across more than one entity level.

Four services that are commonly associated as cross-area services in the TAFIM TRM for example are: Internationalization, Security, System Management and Distributed Computing. These services are identified as such in the document only for the purposes of establishing backwards traceability to the TAFIM TRM. The degree of freedom allowed by the tailoring process provides for the reallocation of certain lower-level services into a functional service area of its own. A good example of this is found in certain object models used by the object-oriented community, in which all of the services associated with the definition and management of objects are collected into a “new” major service area called “Object Services.” Thus, in the Object Model Group’s OMG model, object services associated in such TAFIM TRM categories as Distributed Computing and User Interface Services are found in a single service area called “Object Services.”

4.4.3 External Environment Entity Services

The External Environment Entity represents the external entities with which the application platform exchanges information. These entities are classified in the general categories of User Interface Devices, Persistent Storage, Networks, and Devices. User Interface Devices support physical interaction between the human being and the application platform. Examples of this type of device include CRT displays, keyboards, mice, touch screens, audio input/output devices, and other input and output hardware. Persistent Storage entities include, for example, removable disk packs and floppy disks. Networks include telephone lines, local area networks, cabling, and packet-switching equipment. Devices include hardware interconnect services components and processing resources required to support application platforms (see Section 4.4.2.3.1 for further information on the subject).

4.5 INTERFACE VIEW

The DoD TRM embraces the hierarchical layering and direct and logical interface semantics elaborated in the GOA Framework. The layering, with interfaces between layers and among peers within the layers, is shown in Figure 4.5-1. Functionality within the services view of the DoD TRM is realized in the layering concept. Each layer provides an isolation of the details from other layers. This isolation, using the defined interfaces, enables layer functionality to be portable, reusable, and implementation/technology-independent.

Selection and elaboration of interfaces are dependent on the Program or Development Manager’s application domain and requirements (part of the tailoring process). For example, weapon systems require services and interfaces that support embedded real-time performance and associated detailed interface relationships. The services view of the model focuses on service categories with their composition and relationships; the interface view focuses on interface types and component interconnect relationships. Interfaces establish the “connective links” between entities of the model.

The DoD TRM references the two GOA Framework classes of direct and logical interfaces that are noted in the Figure 4.5-1 diagram. This diagram includes a tailoring of the GOA Layer 4 to provide a mapping of mission and support service applications in the DoD TRM services view. To be consistent with the GOA terminology, this interface has been identified as 4X just as 3X is defined in the third layer in Figure 4.5-1. It is not intended as a modification or change to the GOA Framework, but reflects an elaboration of the structure within the GOA Layer 4 to further define a mission support interface for the application layer.

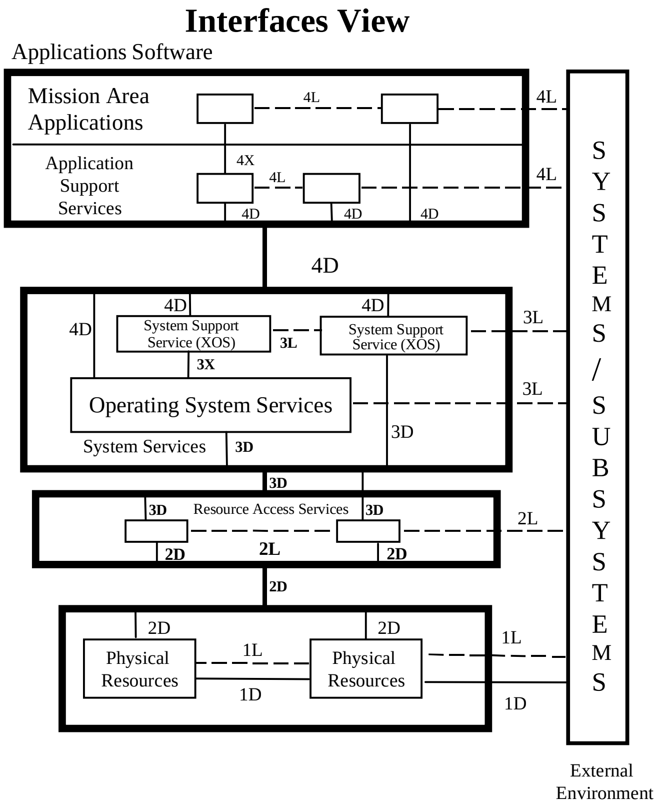


Figure 4.5-1 DoD Technical Reference Model Interfaces View Representation Highlighted

4.5.1 Interfaces View Layers

For a full description of the GOA layers referenced here, refer to the SAE AS4893 document, paragraph 3.1. The section describes the four primary GOA layers. Further information can be found in NASA SGOAA Standard Specification, NASA CR – 188290.

4.5.2 Logical Interfaces

Logical interfaces establish peer-to-peer relationships between components within the same layer of the model and are the horizontal interfaces in Figure 4.5-1. Information routing is transparent to logical interface entities. Logical interfaces are implemented in the GOA model by one or more direct interfaces. These interfaces are fully defined and described in the SAE AS4893 standard and the NASA SGOAA Standard Specification, NASA CR – 188290.

4.5.3 Direct Interfaces

Direct interfaces establish channels of communication between components and are the vertical interfaces in Figure 4.5-1. Information content is typically not of concern in the information routing. These interfaces are fully defined and described in the SAE AS4893 document. Note the following elaboration of GOA Layer 4 internal structure below. This interface was added to facilitate mapping to the services view, which delineates the application software layer into common support and mission-area applications.

- **Class 4X** is a category of direct interfaces between Mission Applications components and Support Services components within the Applications Layer on a given applications platform.

4.5.4 Application Program Interfaces (API)

The API is defined as the interface between the application software and the application platform across which all services are provided. It is defined primarily in support of application portability, but system and application software interoperability is also supported via the communication and the information services segments of the API. The API specifies a complete interface between the application software and the underlying application platform and may be divided into the following groups:

- System Services API (including APIs for Software Engineering Services and Operating System Services).
- Communications Services API (including APIs for Network Services).
- Information Services API (including APIs for Data Management Services and Data Interchange Services).
- Human-Computer Interaction Services API (including APIs for User Interface Services and Graphics Services).

The first API group, System Services, is required to provide access to services associated with the application platform internal resources. The last three API groups (Communications Services, Information Services, and Human-Computer Interaction Services) are required to provide the application software with access to services associated with each of the external environment entities. APIs for services that cut across the areas are included among all groups where applicable.

A standardized API should be used for accessing security mechanisms. The use of the operating-system kernel for maintaining separation among processes executing at different security levels means that this API will be included in the System Services API category above. Such an API will promote independence of security services and security mechanisms, offering transparency to users and applications. This independence will allow different security mechanisms to be accommodated at various stages in an information system life cycle.

4.5.5 External Environment Interfaces

The External Environment Interface (EEI) is the interface between the application software and its platform and the external environment across which information is exchanged. It is defined primarily in support of system and application software interoperability. The EEI includes the interface classes 1D, 1L, 2L, 3L, and 4L. User and data interoperability are directly provided by the EEI. The EEI specifies a complete interface between the application platform and the underlying external environment and includes the following key groups:

- Human-Computer Interaction Services EEI.
- Information Services EEI.
- Communications Services EEI.

The Human-Computer Interaction (HCI) Services EEI is the boundary across which physical interaction between the human being and the application platform takes place. Examples of this type of interface include CRT displays, keyboards, mice, and audio input/output devices. Standardization at this interface will allow users to access the services of compliant systems without costly retraining.

The Information Services EEI defines a boundary across which external, persistent storage service is provided, where only the format and syntax (logical interfaces) are required to be specified for data portability and interoperability.

The Communications Services EEI provides access to services for interaction between application software entities and entities external to the application platform, such as application software entities on other application platforms, external data transport facilities, and devices. The services provided are those in which protocol state, syntax, and format all must be standardized for application interoperability.

Security mechanisms to provide security services in EEIs will be implemented similarly to those required for communications among distributed platforms. That is, the EEIs facilitate communications among distributed platforms.

4.5.6 Interface Types

While establishing an interface and the associated/corresponding standards to support interoperability, it is important to establish which is the entity that provides or receives those interactions or data. The interface view presented identifies classes of interfaces and their association within an entity. They are to be used as a guideline in defining a set of interfaces and subsequently their corresponding or associated standards. The existence of a service or interface type in the model does not imply that it (service or interface) must be used or accounted for in every instance. The extensive list of services and interfaces represents a list that one can select from on an as-needed basis, hence use of the term “tailoring.” It is important, however, that all “critical components” and “critical interfaces” be identified and aligned with the DoD TRM.

4.6 MULTIPLE VIEWS

Figure 4.6-1 represents a composite view showing both services and interfaces together. Under normal circumstances one or the other view would be used in addressing a particular interoperability issue. However, there is nothing that prevents both of these views from being shown together other than the ability of readers to utilize it in their problem domain or to present an issue. Emerging technology demands may in certain systems (e.g., command and control), require a service representation; and the same system in another application scenario may require interface views to support a real-time requirement. The composite views of the DoD TRM illustrate the adaptability and flexibility of the enhanced model presented in this document.

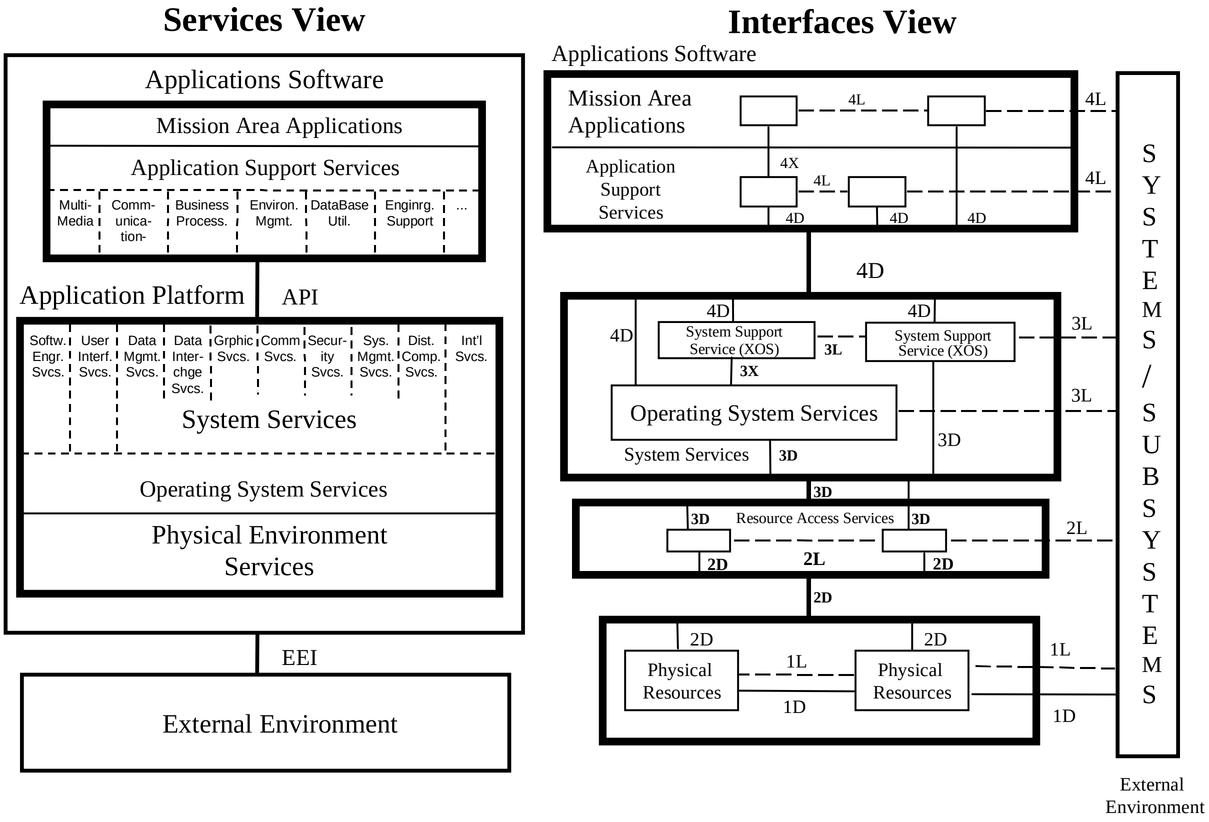


Figure 4.6-1 DoD Technical Reference Model (DoD TRM)

SECTION 5: USING THE TECHNICAL REFERENCE MODEL

5.1 PURPOSE

The Department of Defense (DoD) Technical Reference Model (TRM) is intended to be used by anyone involved in developing systems that must exchange information or interoperate with other systems (e.g., developers, system architects, customers). The latter pertains to systems that exchange information or are required to use the information directly or indirectly. The model is to be used to develop technical architectures and as an aid in defining interoperability relationships and parameters.

The purpose of the DoD TRM is to provide a common conceptual framework and a common vocabulary (i.e., services, interfaces, entity relationships) to assist in the identification of interoperability relationships and resolution of open-systems issues. It is not an architecture nor does it provide one.

The model is most effectively used to define an interoperability or open-system framework <u>in conjunction</u> with the following:

- A set of identified user or system requirements.
- A set of identified user functional relationships.
- A requirements methodology.
- A technical architecture.
- A systems architecture.

TAFIM Volume 4, Standards-based Architecture Planning Guide, provides valuable insight in developing an architecture methodology. Additionally, Services and Agency Components may employ their own requirements or functional capabilities methodology.

The model is a tool to be used to define consistent and common terms between those systems required to interoperate or those systems encountered within an enterprise, a battlefield functional area, or functional domain. The model is tailorable to accommodate a variety of DoD requirements. Tailoring is the process of using a model and adjusting it to fit specific domain or implementation requirements.

5.2 GUIDANCE

Proper development of a systems architecture is dependent on establishing a technical and operational architecture. Proper definition and development of a system depends on a well-defined set of user and functional requirements. Reconciling the systems development process with the architectural process requires several transformations of requirements to functions and capabilities, and thence functions to services or interfaces that support those functions and requirements. Subsequently, standards are mapped to those same services/interfaces to aid in the process of selecting an appropriate set of standards in support of the systems requirements. Further elaboration of mappings and mapping transformations are deferred to other documents on the subject. Requirements, functional mappings, and standards assessment methodologies are beyond the scope of this document and are not addressed any further.

Use of the DoD TRM to aid in the selection of a common set of services and interfaces is very important when supporting interoperability. The following additional comments on DoD TRM usage are presented:

- The two views (i.e., services and interfaces) are not required to be used concurrently unless required by the user. Interoperability issues are driven by system requirements, operational domain, and information exchanged between systems in question. The selection of a particular model view is driven by these same requirements and, similarly, by the information exchanged. Thus, only those services or interfaces identified by the systems analysis process are required to support the technical architecture. The set of DoD TRM model services and interfaces present a more exhaustive set than those required in actuality (i.e., they need not all be used because they are in the model).
- In certain instances a high level service definition (e.g., network services) may be appropriate, but a related lower level service elaboration (e.g., Web support service) is found to be inadequate or non-existent. In these cases, TRM users are encouraged to define their lower level service for purposes of clarification and to submit them to the TRMWG for future incorporation into subsequent TRM versions. Similarly, new technologies may require new service definitions to support them that are not currently in the model.
- Real-time and performance requirements may require direct interfaces between entity levels that are not ordered or in sequence (i.e., require bypassing an entity or sub-entity layer). The DoD TRM interface views accommodate this type of connectivity (e.g., null services or bypassing layers) to support real-time or performance needs (although this practice is not recommended). System performance requirements may dictate direct connectivity between a mission application support layer element and a resource driver, as in the case of some sensors.
- Systems within an operational domain, that are required to interoperate, do not all have to utilize the same set of services. However, when two systems require the same service, use of the same service definition is required.
- The set of services and interfaces identified in the model should be considered an initial set, and not the definitive (all-inclusive) set. This facet provides for model evolution

and inclusion of new features as warranted.

- The model may be duplicated (see Figure 4.3.4-1) to support multiplatform configurations and the association of allocated services and interfaces for each platform.

In defining an interface, the following guidelines are provided to help select the appropriate component and relationship:

- All physically connected entities/components are to have clearly defined inputs and outputs. The interfaces between these two entities are defined as direct interfaces.
- All entities that exchange information through the use of intermediate entities are to be considered interoperable (to some degree)[1], provided that the intermediate systems do not process the essential information other than for format or protocol considerations. The interface between these two entities will be defined as logical. In contrast to direct interfaces, logical interfaces are dependent on third-party entities or systems.
- For every logical interface there should be a corresponding (at least one) direct interface(s). An individual system can have direct interfaces without a corresponding logical interface, but a logical interface can only be achieved through direct interfaces between multiple systems.
- For intermediate entities providing value-added processing to the information being transferred, the interaction should be considered as occurring with the closest intermediate entity providing such processing (the interaction could be direct or logical).
- Interfaces to external entities not electronically connected should be considered interoperable if they are essential to the processing of the mission. Those interfaces could be non-direct (e.g., transfers of physical media, human-in-the-loop interactions), or shared resource (the data is accessed through common information repositories). It will be up to the discretion of the individual Program Manager or System Architect to define which of these interfaces will be considered mission essential and supportive of user needs. The interfaces could be considered direct or logical, depending on whether there are intermediate systems and their processing.

5.3 NEW-SYSTEM DEVELOPMENT

The DoD TRM can be used to support new-system developments as well as legacy system and system migration efforts. In a new-system development, the initial task is to define the operational scenario (i.e., system operates in the environment as a stand-alone system, a point system; or system interoperates with other systems). Once the operational scenario is established, a model methodology can be developed or used. A point-system scenario does not require elaborate cross-checking of services or interfaces with other reference models to ensure consistency of terms or equivalency of model structure and relationships. However, a scenario with interoperability requirements requires the following:

[1] The LISI model (see Appendix A) should be used to assess the degree of interoperability.

- Comparison of reference models used by the systems that need to interoperate (i.e., views presented – services, interfaces, other).
- Comparison of models for equivalency of structure and entity relationships.
- Comparison of model for equivalency of service and interface definitions.
- Identification of specific services and interfaces required in the new system and to support interoperability.
- Comparison of service and interface definitions between the system under development and the other related systems in the operational environment to ensure consistency and commonality.

5.4 EXISTING SYSTEM ASSESSMENT

In using the model to assess an existing system, identifying the operational scenario follows the same basic approach as that of Section 5.3, with several additional items for consideration:

- The degree of openness of the existing system must be evaluated relative to the operational scenario, both existing and future.
- Differences identified in the first bullet are then used as drivers for developing an effective migration strategy.
- Focus on commonality opportunities or elements with those of the other systems in the same environment. Commonality opportunities (i.e., they provide common denominators) serve as the basis for the initial migration increments and for establishing a baseline to migrate to.

Examples of DoD TRM usage are presented in Appendix E.

APPENDIX A - REFERENCES

1. IEEE Guide to the POSIX Open System Environment (IEEE 1003.0-1995), Institute of Electrical and Electronics Engineers, Inc., 1995.

2. NIST, FIPS 146-2, Profiles for Open Systems Technologies, May 1995.

3. NIST Special Report 500-187, Application Portability Profile (APP): The U.S. Government's Open System Environment Profile OSE/1, Version 1.0, 27 October 1992.

4. NIST Special Publication 500-163, Government Open Systems Interconnection Profile (GOSIP) User's Guide, 2nd Edition May 1991.

5. NIST Special Publication 500-201, Reference Model for Frameworks of Software Engineering Environments (Technical Report ECMA TR/55, 2nd Edition), December 1991.

6. OSF, OSF/Motif Application Environment Specification User Environment, Volume 1.0, Rev A, 1990.

7. OSF, OSF/Motif Programmer's Guide, Rev 1.0, 1990.

8. OSF, OSF/Motif Style Guide, Rev 1.0, 1990.

9. OSF, OSF/Motif User's Guide, Rev 1.0, 1990.

10. Plan for Implementation of Corporate Information Management in DoD, ASD/C3I, 8 January 1991.

11. SECDEF Memo, 16 November 1990, Implementation of Corporate Information Management Principles w/Enclosure.

12. Sun Microsystems, Inc., Open Look Graphical User Interface Application Style Guidelines, 1989.

13. Sun Microsystems, Inc., Open Look Graphical User Interface Functional Specifications, 1989.

14. Strategies for Open Systems, Stage Two: The Experience With Open Systems, DMR Group, Inc., Boston, 1990, pp. 196.

15. X/OPEN Company, Ltd., X/OPEN Portability Guide, Version 3 (XPG3), 1988.

16. X/Open Portability Guide, Issue 3, Volumes 1-7, X/Open Company, Ltd., Prentice Hall, Inglewood Cliffs, NJ, 1988.

17. Department of Defense, Levels of Information Systems Interoperability (LISI), C4ISR Architecture Working Group, 30 March 1998.

18. Department of Defense, C4ISR Architecture Framework, Version 2.0, C4ISR Architecture Working Group, 18 December 1997.

19. Defense Information Infrastructure, Master Plan, Version 7.0, Defense Information Systems Agency, 13 March 1998.

20. Defense Information Infrastructure (DII), Common Operating Environment (COE), Integration and Runtime Specification (IandRTS), Volume 3.0 (DRAFT), Defense Information Systems Agency, January 1997.

21. Department of Defense, Technical Architecture Framework for Information Management (TAFIM), Version 3.0, Defense Information Systems Agency – Center for Standards, 30 April 1996, Volume 1 through 8.

22. Department of Defense, Joint Technical Architecture, Version 2.0, 26 May 1998.

23. SAE 4893, Generic Open Architecture (GOA) Framework, SAE AS-5 GOA Task Group, January 1996

24. ITSG, DISA Center for Standards, Information Technology Standards Guidance, 14 Volumes, December 1997.

25. Defense Information Technology Security Certification Accreditation Procedure (DITSCAP), 30 December 1997.

26. Joint Vision 2010, CJCS, September 1996.

APPENDIX B - DIRECTIVES AND MEMORANDA

Appendix B contains Secretarial Letters referencing the subject of Technical Reference Models (explicit and implicit) and related subjects. The Letters are indicative of the important role a technical reference model plays in defining an architecture. The contents of this Appendix also provide historical context to the DoD TRM presented in this document.

June 18, 1997

M-97-16

MEMORANDUM FOR THE HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES

FROM: Franklin D. Raines

SUBJECT: Information Technology Architectures

This memorandum transmits guidance to Federal agencies on the development and implementation of Information Technology Architectures. The Information Technology Architecture (ITA) describes the relationships among the work the agency does, the information the agency uses, and the information technology that the agency needs. It includes standards that guide the design of new systems. An ITA makes it easier to share information internally (e.g., agency-wide e-mail) and to reduce the number of information systems that perform similar functions. The ITA provides the technology vision to guide resource decisions that reduce costs and improve mission performance.

OMB Memorandum 97-02, "Funding Information Systems Investments," (October 25, 1996), requires that agency investments in major information systems should be consistent with Federal, agency, and bureau ITAs. The Clinger-Cohen Act of 1996 (Public Law 104-106) assigns the Chief Information Officer the responsibility of developing, maintaining, and facilitating the implementation of the information technology architecture.

As described in the attachment, a complete ITA is the documentation of the relationships among business and management processes and information technology that ensure:

alignment of the requirements for agency-sponsored information systems (as defined in OMB Circular A-130) with the processes that support the agency's missions and goals;

adequate interoperability, redundancy, and security of information systems; and,

the application and maintenance of a collection of standards by which the agency evaluates and acquires new systems.

Agencies should be prepared to indicate the status of the development, implementation, and maintenance of the agency ITA during the formulation of the FY 1999 President's budget.

Attachment

Development, Maintenance, and Implementation of Agency Information Technology Architectures

Table of Contents

Purpose

The purpose of this paper is to establish minimum criteria for an agency information technology architecture (ITA) required in the Clinger-Cohen Act of 1996 (Public Law 104-106).

Background

The Clinger-Cohen Act assigns the Chief Information Officers (CIO) the responsibility of "developing, maintaining, and facilitating the implementation of a sound and integrated information technology architecture." (Section 5125 (b) (2)) The Act defines the ITA as:

> an *integrated framework* for evolving or maintaining existing information technology and acquiring new information technology *to achieve the agency's strategic goals* and information resources management goals. (Section 5125 (d)) (Emphasis added.)

OMB's memorandum 97-02, "Funding Information Systems Investments," dated October 25. 1996, states,

> Investments in major information systems proposed for funding in the President's budget should be consistent with Federal, agency, and bureau information architectures which: integrate agency work processes and information flows with technology to achieve the agency's strategic goals; and specify standards that enable information exchange and resource sharing.

These references highlight three important characteristics of the ITA as agencies plan for investments in information technology (IT) assets:

- CIOs are responsible for the architecture;
- the architecture must integrate the business processes and goals of the agency with IT acquisitions; and,
- the architecture focuses on work processes, information flows, and standards.

Agencies may address the topics and elements set out herein in a manner appropriate to the agency. Each element identified need not have specific or "stand-alone" documentation.

Information Technology Architecture Defined

For the purpose of conforming to the requirements of Clinger-Cohen Act, a complete ITA is the documentation of the relationships among business and management processes and information technology that ensure:

- alignment of the requirements for information systems (as defined in OMB Circular A-130) with the processes that support the agency's missions;
- adequate interoperability, redundancy, and security of information systems; and,
- the application and maintenance of a collection of standards (including technical standards) by which the agency evaluates and acquires new systems.

Developing the ITA

The ITA is broad in scope and includes processes and products. An architecture in compliance with the Clinger-Cohen Act and OMB guidance will contain two elements:

- the Enterprise Architecture,
- a Technical Reference Model and Standards Profiles.

In developing their ITAs, agencies are *not* required to use the terminology contained in this guidance. Examples of various agency architectures may be found in Appendix I.

A variety of nomenclatures are available to address these elements. Agencies may address the elements of an ITA in different ways and at various levels of granularity as appropriate, combining or reorganizing the parts to create a model that suits the agency's organizational needs. Various aspects of the ITA can be developed at the agency or sub-agency level. However, self-contained sub-agency level architectures should be integrated and consistent with an agency-wide ITA.

The Enterprise Architecture

The Enterprise Architecture is the explicit description of the current and desired relationships among business and management process and information technology. It describes the "target" situation which the agency wishes to create and maintain by managing its IT portfolio.

The documentation of the Enterprise Architecture should include a discussion of principles and goals.[1] For example, the agency's overall management environment, including the balance between centralization and decentralization and the pace of change within the agency, should be clearly understood when developing the Enterprise Architecture. Within that environment, principles and goals set direction on such issues as the promotion of interoperability, open systems, public access, end-user satisfaction, and security.

This guidance adapts a five component model used in the National Institute of Standards and Technology (NIST) Special Publication 500-167, "Information Management Directions: The Integration Challenge." Agencies are permitted to identify different components as appropriate and to specify the organizational level at which specific aspects of the components will be implemented. Although the substance of these components, sometimes called "architectures" or "sub-architectures,"[2] must be addressed in every agency's complete Enterprise Architecture, agencies have great flexibility in describing, combining, and renaming the components, which consist of:

- Business Processes
- Information Flows and Relationships
- Applications
- Data Descriptions
- Technology Infrastructure

With the exception of the Business Processes component, the interrelationships among and priorities of these components are not prescribed by this guidance; there is no hierarchy of relationships implied. Furthermore, agencies should document not only their current environment for each of these components, but also the target environment that is desired.

Business Processes

This component of the Enterprise Architecture describes the core business processes which support the organization's missions. The Business Processes component is a high-level analysis of the work the agency performs to support the organization's mission, vision, and goals, and is the foundation of the ITA. Analysis of the business processes determine the information needed and processed by the agency. This aspect of the ITA must be developed by senior program managers in conjunction with IT managers. Without a thorough understanding of its business processes and their relation to the agency missions, the agency will not be able to use its ITA effectively.

Business processes can be described by decomposing the processes into derivative business activities. There are a number of methodologies and related tools available to help agencies decompose processes. Irrespective of the tool used, the model should remain at a high enough level to allow a broad agency focus, yet sufficiently detailed to be useful in decision-making as the agency identifies its information needs. Agencies should avoid excessive emphasis on modeling business processes, which can result in a waste of agency resources.[3]

Information Flows and Relationships

This component analyzes the information utilized by the organization in its business processes, identifying the information used and the movement of the information within the agency. The relationships among the various flows of information are described in this component. These information flows indicate where the information is needed and how the information is shared to support mission functions.[4]

Applications

The Applications component identifies, defines, and organizes the activities that capture, manipulate, and manage the business information to support *mission* operations. It also describes the logical dependencies and relationships among business activities.[5]

Data Descriptions and Relationships

This component of the Enterprise Architecture identifies how data is maintained, accessed, and used. At a high level, agencies define the data and describe the relationships among data elements used in the agency's information systems. The Data Descriptions and Relationships component can include data models that describe the data underlying the business and information needs of the agency. Clearly representing the data and data relationships is important for identifying data that can be shared corporately, for minimizing redundancy, and for supporting new applications.[6]

Technology Infrastructure

The Technology Infrastructure component describes and identifies the physical layer including, the functional characteristics, capabilities, and interconnections of the hardware, software, and communications, including networks, protocols, and nodes. It is the "wiring diagram" of the physical IT infrastructure.[7]

Technical Reference Model and Standards Profiles

The Technical Reference Model (TRM) and Standards Profiles (both technical and security) comprise a cross-cutting element, affecting all components of the Enterprise Architecture. Standards enable interoperability, portability, and scaleability in systems throughout the agency. Although the specificity of the standards may vary by organizational level, the standards must be consistent throughout the agency. Standards are the basis of the development of components of the Enterprise Architecture and ultimately guide and constrain IT asset acquisitions.

Technical Reference Model

The TRM identifies and describes the information services (such as database, communications, and security services) used throughout the agency. For example, Data Interchange Services support the exchange of data and information between applications. This information service would identify the various ways the agency enables the exchange of data, such as plain text, spreadsheets, databases, graphical information over Intranet/Internet, and video.

Standards Profiles

The standards profile defines a set of IT standards that supports the services articulated in the TRM; they are the cornerstone of interoperablity. The profile establishes the minimum criteria needed to specify technology that achieves the purposes of standardization and supports specific business functions. Standards Profiles are the published sets of standards or the source references for standards that prescribe the interfaces between those services that will be standards-based. A profile may contain specifications that describes the technical standards which enable a service, such as operating systems, network, and data interchange services.

Together with the TRM, the Standards Profiles enable the development and acquisition of standardized systems to cost-effectively meet the business needs of the agency. Agencies are expected to adopt the minimum standards necessary to support all components of the desired Enterprise Architecture. The profiles should address hardware, software, communications, data management, user interfaces, and implementation approaches, and may indicate specific products that implement the standard.[8]

Security Standards Profiles

While security services may be considered part of the TRM and security profiles may be a subset of the standards profiles, the importance of security as a cross-cutting issue warrants special attention. Security standards need not be a separate component of the Enterprise Architecture or of the TRM. Security standards profiles are Standards Profiles specific to the security services specified in the Enterprise Architecture. The profiles cover such services as: identification, authentication, and non-repudiation; audit trail creation and analysis; access controls; cryptography management; virus prevention; fraud prevention, detection and mitigation; and intrusion, prevention and detection. The purpose of the security profiles is to establish information and technology security standards to ensure adequate security for each component of the Enterprise Architecture, and to ensure that information systems conform to agency security policy. The security standards identified in the security standards profiles must be consistent with the requirements of OMB Circular A-130, Appendix III.

Maintaining and Implementing the ITA

While the development of an Enterprise Architecture is important, only its successful implementation meets the goals of the Clinger-Cohen Act. Having established a framework for the ITA, each agency should prioritize areas of high incremental benefits for early implementation. Particular attention should be given to the following:

- Change Management
- Legacy Systems Integration
- IT Personnel Planning
- Compliance, Waivers, and Certification

Change Management

Developing an ITA is an iterative and dynamic process. The ITA should be revised periodically so that it evolves as the agency's business functions evolve. Thus, the ITA itself should be managed with the same change control process that governs other critical documents.

A baseline of the current environment -- how and where IT assets are currently used -- should be part of the initial development of the ITA, and the baseline should be maintained over time. The ITA should reflect the agency's technology research effort. Every agency should have a mechanism for evaluating current technologies and identifying new IT opportunities for the agency. An option for many agencies may be the establishment of an ITA board to act as a steward of the ITA and to perform these ITA development and maintenance activities.

Legacy Systems Integration

A useful ITA must realistically account for the existing infrastructure base, including legacy systems. In this context, "legacy systems" refers to systems currently in use. The architectural strategy for dealing with legacy systems should focus on their interfaces with new systems, permitting the new and the old to interoperate in a cost-effective manner that does not compromise the ability of the new system to conform completely with the target architecture and standards. If the user interface of an older system does not conform to the architecture, a decision whether to change, replace, or terminate will turn on cost, operational, or functional effectiveness criteria.

Information Technology Personnel Planning

The ITA should reflect the training, procedures, and staffing needed to support its successful implementation. Agencies should identify the human resources and technical skills needed and available to develop, maintain, and implement the ITA. Agencies should plan for the remediation of deficiencies, including strategies and plans for hiring, training, and professional development (Clinger-Cohen, Section 5125 (c) (3)).

ITA Compliance, Waivers, and Certification

The ITA itself should guide systems changes for new and operational systems. Conformance to the ITA and compliance with the standards profiles is critical to success. Configuration management and control as well as quality software engineering processes for systems should be implemented to maintain compliance with the architecture. Configuration changes should be tested and validated prior to acceptance for operational use across the architecture.

To migrate from the agency's current environment to the target architecture, new systems will increasingly have to meet the standards of the ITA. The ITA should not be weakened nor should its impact as a tool diluted through the excessive use of waivers. The CIO and other senior managers should require strong business case justifications for exceptions to the ITA. Waivers for non-conformance should be the exception, not the rule, and waivers should only be granted after a careful, thorough, and well documented analysis which supports the need for the exception.

An ITA should have an established method of evaluating the level of compliance of proposed new systems and of proposed modifications to current systems. This method may be formalized to the point of a certification process. At a minimum, metrics should be established which, if met, permit a proposed system to be termed "ITA compliant."

Appendix I: Published Architecture Model Sources

DEPARTMENT OF AGRICULTURE

The U.S. Department of Agriculture (USDA) has recently developed its first version of the USDA Information Systems Technology Architecture (ISTA). The Architecture is a high-level document divided into three components: Business and Data Architecture (Part I), Technical Standards Architecture (Part II), and Telecommunications Architecture (Part III). The Business and Data Architecture identifies core business processes for each of USDA's mission areas and associated common data elements. The Technical Standards Architecture has three tiers: Tier I "Core Technologies," Tier II "General Purpose Productivity Enabling Technologies," and Tier III "Integrating Technologies." The Telecommunications Architecture identifies an enterprise network architecture for USDA.

The three components of the USDA ISTA can be mapped to the NIST model. The Business/Data Architecture aligns with the Business Functions and Data Descriptions layers, and the Technical Standards and Telecommunications Architectures align with the Technology Infrastructure layer. The USDA ISTA is a living document and will be continually refreshed to ensure that USDA employs established and emerging technology to meet its strategic business goals.

The Office of the Chief Information Officer has also developed a set of criteria to guide the agency’s IT investment decisions based on OMB Memorandum 97-02. USDA is establishing the required management mechanisms and tools to ensure successful integration and implementation, assessment, and monitoring of the USDA architecture needs. Additional information and copies of the USDA ISTA may be obtained from Mr. Joseph Ware, Chief, Information Management Division, Office of the Chief Information Officer, USDA; phone: (202) 690-2118; fax: (202) 690-2831; or E-mail: joe.ware@usda.gov.

DEPARTMENT OF DEFENSE

The C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance) Architecture Framework, Version 1.0 , prepared by the C4I Integration Support Activity (CISA), serves as guidance to Department of Defense (DoD) organizations that need to develop architectural descriptions for their internal use or to support broader Departmental activities. The objective of the Framework Version 1.0 is to provide guidelines for developing architecture descriptions that are internally consistent within themselves, of practical use to decision makers at all appropriate levels, and will integrate with other architecture descriptions across DoD.

The C4ISR Architecture Framework is organized by the three architecture "views": Operational, Systems, and Technical. The framework details products that may be needed in the course of describing an architecture view and also indicates the kinds of information to be captured in each product. While the Operational, Systems, and Technical Architectures frequently are discussed as if they were separate architectures, they are best considered as different views of an architecture -- each focusing on particular aspects. The three architectures views are defined as follows:

1. Operational Architectures containing "descriptions of the tasks, operational elements, and information flows required to accomplish or support a warfighting function." i.e., the pertinent activities, operational elements, and associated information flows.

2. Systems Architectures describing "systems and interconnections which support the warfighting functions,"i.e., systems (including Automated Information Systems, communications, and weapon platforms) used to satisfy operational needs and the corresponding interconnections.

3. Technical Architectures are "a minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements whose purpose is to ensure that a conformant system satisfies a specified set of requirements," such as, technical standards, criteria, and reference models that govern the implementation of systems to satisfy the operational needs.

The aforementioned architectures are described by products that are graphical, database, and/or textual. For convenience, the products are categorized according to their main view: Operational, Systems, or Technical. However, any given architecture description will consist of the best combination of products to illuminate the issue area, whether they are categorized as Operational, Systems, and/or Technical products. In addition to the view-specific products, Information Infrastructure Products provide support within and/or across the views. These products include entity-relationship models, attributed information models, and an Integrated Data Dictionary.

For more information about the C4ISR Architecture Framework, Version 1.0, please contact Mr. Jim Bain, Director of the Architectures Directorate in the C4I Integration Support Activity, at 703-883-6907 or by E-mail at james.bain@osd.mil.

DEPARTMENT OF ENERGY

The Department of Energy's Information Architecture (IA) is being defined in four volumes, as follows:

Volume I, "The Foundations," issued in March 1995 - Prescribes the IA Principles, the Conceptual IA Model, a high level business case, the current IA baseline, the standards process, a summarized vision, IA policy, and next steps.

Volume II, "The Baseline Analysis," issued in December

1996, is a three part document consisting of the Baseline Analysis Summary (Part 1), the Detailed Baseline Analysis (Part 2), and Baseline Analysis Reference (Part 3). The three parts are founded on an extensive Department-wide analysis using a series of models to assess and extract the significant challenges of the IA. These challenges are described in key areas and are graphically depicted. Findings and conclusions are also presented.

Volume III, "Guidance," targeted for April 1997 (currently in Draft), provides specific guidance to Departmental elements, strongly recommending the formalization of Information Architectures at various levels within DOE (particularly for Programs and sites). The IA principles, conceptual model, minimal design characteristics, IA program, and standards are covered.

Volume IV, "IA Vision," targeted for September 1997, defines the future Departmental Architecture by (subarchitecture) layer, depicting the targeted capabilities and structures envisioned by function. It will give specific examples of business functions and how they are depicted in each layer. It will also relate the standards to each applicable layer.

Other documents available include: "The Standards Service Action Plan," (March 1997), the "DOE IA Standards Profile," the "Standards Process Guide," and the "IA Methodology Guide."

In general, DOE is following the NIST five-layer architectural model, because it ensures that linkages are established from the business functions and processes down through the information, applications and data layers, down to the specific technologies utilized to support the business. The above documents go into detailed explanations regarding what each layer addresses and on other facets of DOE’s IA Program.

These documents can be obtained on the DOE IA Web site at HTTP://www-it.hr.doe.gov/iat/, or by contacting Mr. Michael Tiemann, DOE IA Project Manager at michael.tiemann@hq.doe.gov or (202) 586-5461.

DEPARTMENT OF TREASURY

The Treasury Information System Architecture Framework (TISAF) is a model to assist Treasury Bureaus to develop their Enterprise Information System Architectures (EISAs). The TISAF consists of a list of goals and objectives for planning Treasury information technology, a set of architectural principles for developing information systems, an EISA model for describing distinct views of enterprise information systems, and a set of standards for guiding specific product selection. The EISA model provides four architectural views to organize, plan, and build enterprise information systems, consisting of the Information, Functional, and Work architectures and the Infrastructure.

The Information Architecture is the "what" of information Systems, which defines and organizes all information needed to perform business operations and describes the relationships among this information. The Functional Architecture is the "how" of information systems, which defines and organizes the business functions, processes, or activities that capture, manipulate, and manage the business information to support business operations. The Work Architecture is the "where" of information Systems, which depicts the decentralization of the business, the description of the work organizations to business locations, and the communications and coordination between these locations. The Infrastructure is the "enabler" of information systems, which describes the supporting services, computing platforms, and internal and external interfaces needed to provide technology environments within which information systems run.

To provide a context for discussing technical standards, a Technical Reference Model (TRM) is developed to organize and depict building blocks of an information system as a set of services categorized by functional areas. For more information and a copy of the TISAF, please contact Mr. Simon Liu at (202) 622-9089, or E-mail at simon.liu@cio.treas.gov.

Appendix II: Additional References

"Application Portability Profile (APP): The U.S. Government’s Open System Environment Profile Version 3.0," Computer Systems Technology, NIST, February 1996.

"Architecture Concepts and Design Guidance (TAFIM)," Department of Defense, vol. 3, ver. 2.0, June 1994.

"C4ISR Architecture Framework," C4ISR ITRF, Integrated Architecture Panel, v. 1.0, Department of Defense, CISA-0000-100-96, June 1996.

"Developing the Information Systems Architecture for World-Class Organizations," Lee, Management Decisions, 34/2, 1996, pp. 46-52.

"Enterprise," Department of the Army Technical Architecture, ver. 4.5, 12 November 1996.

"Experiences and Examples in Development of Information Systems Architecture," Performance Engineering Corporation, Presentation to the Department of Justice, April 1996.

"How to Align Corporate Goals and Information Technology," Dietrich, Communication News, Vol. 32, No. 10, 1 October 1995.

"Information Architectural Design in Business Process Reengineering," Kettinger, Journal of Information Technology, Vol. 11, pp. 27-37, 1996.

"Information Architecture Volume I: The Foundations," Department of Energy, March 1995.

"Information Architecture Volume II: Baseline Analysis Summary" Department of Energy,December 1996.

"Information Architecture Volume III: Guidance," Department of Energy, April 1997.

"Information Management Directions: The Integration Challenge," Computer Systems Technology, National Institute of Standards and Technology, September 1989.

"Information Systems Technology Architecture Review," Information Technology Resources Board (ITRB), December 1996.

"Joint Technical Architecture," C4ISR, Department of Defense, vol. 1.-0, August 1996.

"Open System Environment (OSE): Architectural Framework for Information Infrastructure," Schulz, NIST Special Publication 500-232, September, 1995.

"Levels of Information System Interoperability," for the C4I Integration Support Activity, Architecture Directorate, MITRE Corporation, June 1996.

"Perspectives," Hurwitz, Computer World, 1 October 1995.

"Plan Your Top Priorities," Cash, Information Week, 4 March 1996.

"Standards-Based Architecture (SBA) Planning Guide," Defense Information Systems Agency (DISA), October 1993.

"A Systems Engineering Approach to Information Architecture Design," Levis, IFAC Integrated Systems Engineering, 1994, pp. 131-144.

"Technical Architecture Framework for Information Management (TAFIM)," Department of Defense, Vol. 1: Overview, ver. 2.0, June 1994.

"Technology Defiant: Your Ticket To Ride?" Braue, Data Communications, Vol. 24. No. 14, 1 October 1995.

"Treasury Information Systems Architecture Framework," Department of Treasury, ver. 1.0, January 1997.

"What is an Information Technology Architecture," IDC Government, January 1996.

*****Footnotes

[1.] Examples of published architectural "frameworks" include the Treasury Information System Architecture Framework (TISAF), the Department of Defense Technical Architecture Framework for Information Management (TAFIM), and the Department of Energy’s Information Architecture
Volume 1.

[2.] Examples of agency efforts to develop Enterprise Architectures and how the agencies have named and described these components are found in Appendix I.

[3.] The Department of Defense includes aspects of the Business Processes element in its Operational Architecture; the Department of Treasury incorporates it into its business view.

[4.] The Department of Agriculture has incorporated this component into its Business Architecture, while the Department of Defense and Treasury have built it into their Operational Architectures.

[5.] The Department of Energy incorporates Business Applications into its Applications Subarchitecture, while the Department of Treasury includes them in its Functional Architecture.

[6.] The Department of Agriculture has included this element in its Business/Data Architecture, while the Department of Treasury incorporates it in its Information Architecture.

[7] The Department of Agriculture has incorporated this architecture into its Technical Standard and Telecommunications Architectures. DoD uses its System Architecture, and Treasury its Infrastrucsture to describe the physical layer.

[8] For services not covered by published standards, agencies should identify *de facto* industry standards or specific products that best accommodate an open-system environment.

OFFICE OF THE ASSISTANT SECRETARY OF DEFENSE
3300 DEFENSE PENTAGON
WASHINGTON, DC 20301-3300

C3I 23 June 1994

MEMORANDUM FOR UNDER SECRETARIES OF DEFENSE
COMPTROLLER OF THE DEPARTMENT OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY (RD and A)
ASSISTANT SECRETARY OF THE NAVY (RD and A)
ASSISTANT SECRETARY OF THE AIR FORCE
((ACQUISITION)(SAF/AQ))
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM)

This memorandum affirms Department of Defense (DoD) commitment to the Technical Architecture Framework for Information Management (TAFIM). Since January 1993, the TAFIM has served as the single framework to promote the integration of DoD information systems, thus expanding the opportunities for interpretability and enhancing our capability to manage information resources across the Department. The TAFIM will guide the evolution of the Department's information system technical architectures.

As a long-range goal, the Department is fully committed to an open systems environment, enabling information systems to be developed, operated, and maintained independent of proprietary technical solutions. The TAFIM establishes the direction for an open systems environment that focuses on a standards-based architecture critical to achieving interoperability and cross-functional integration.

New DoD information systems development and modernization programs will conform to the TAFIM, Volume 1 – Implementation Concept, Volume 2 -Architecture Guidance and Design Concepts, Volume 3 -Reference Model and Standards Profile and subsequent forthcoming volumes. The selection and evaluation of migration systems should take into account our long-range goal by striving for conformance to the TAFIM to the extent possible. As stated in my November 12, 1993 memorandum, "Selection of Migration Systems," conformance to the TAFIM is a key technical factor to be considered in the selection of migration systems. In addition, the evolutionary changes to migration systems will be governed by conformance to the TAFIM. Requests for exceptions, with supporting rationale, should be forwarded to this office for consideration.

The TAFIM is maintained by Defense Information systems Agency's (DISA's) Center for Architecture and is available through the National Technical Information Service (NTIS) and the Defense Technical Information Center (DTIC). Attached are the DTIC accession numbers for each document. The TAFIM is an evolving set of documents and comments for improving may be provided to DISA at any time. The action officer for this matter is Mr. Terry Hagle, (703) 604-1486.

s-Emmett Paige, Jr.

(Attachment)

OFFICE OF THE UNDER SECRETARY OF DEFENSE

3300 DEFENSE PENTAGON
WASHINGTON, DC 20301-3300

C3I 30 November 1998

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
DIRECTOR, DEFENSE RESEARCH AND ENGINEERING
ASSISTANT SECRETARIES OF DEFENSE
GENERAL COUNSEL OF THE DEPARTMENT OF DEFENSE
DIRECTOR, OPERATIONAL TEST AND EVALUATION
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTORS OF THE DEFENSE FIELD ACTIVITIES

SUBJECT: DoD Joint Technical Architecture (JTA) Version 2.0

JTA Version 2.0 was approved by the Architecture Coordination Council on May 28, 1998 and has been posted to the JTA Home Page with a notice that formal authorization for use will be provided separately. This memorandum makes JTA Version 2.0 effective for use immediately, superseding version 1.0. In addition, this memorandum updates the portion of Paragraph 4.3.9 of DoD 5000.2-R (with Change 3) covering the JTA applicability and waiver process, pending a formal revision of DoD 5000.2-R and other DoD Directives and Instructions.

Implementation of JTA, that is the use of applicable JTA mandated standards, is required for all emerging, or changes to an existing capability that produces, uses, or exchanges information in any form electronically; crosses a functional or DoD Component boundary; and gives the warfighter or DoD decision maker an operational capability. Use of an applicable JTA mandated standard must consider the cost, schedule, or performance impacts, and if warranted a waiver from use granted as described below. Hence, implementation of the JTA is required for all DoD Acquisition Categories, and all other non-traditional (e.g., Defense Information Infrastructure (DII) Common Operating Environment (COE)), systemic (e.g., Joint Airborne SIGINT Architecture (JASA)), or non-DoD 5000 series acquisitions (e.g., procurement of information technology services, CINC Initiatives) that meet these criteria. In addition, implementation of the JTA is required for pre-acquisition programs such as: Advanced Concept Technology Demonstration (ACTDs), Advanced Technology Demonstrations (ATDs), Joint Warrior Interoperability Demonstrations (JWIDs), 'Exploitation-year', and Battle Laboratory projects that meet these criteria.

Each DoD Component and cognizant OSD authority is responsible for implementation to include compliance assurance, programming and budgeting of resources, and scheduling. Only the Component Acquisition Executive, or cognizant OSD authority can grant a waiver from the use of an applicable JTA mandated standard. All waivers shall be submitted to the USD (A&T) and ASD (C3I) (the DoD Chief Information Officer (CIO)) for concurrence. Both USD (A&T) and ASD (C3I) (DoD CIO) concurrence can be assumed if no response is received two weeks after the date of receipt. To assure proper and timely consideration, all waivers must be accompanied by the identification of cost, schedule, and performance impacts that will occur if waiver is not granted and acknowledgment of any resulting operational limitations.

To preclude the granting of duplicative waivers, caused by implementing this and other OSD mandates, the organization responsible for systemic implementations of the JTA (e.g., DISA for the DII COE; NSA for the JASA; BMDO for the standards in the Missile Defense) will administratively coordinate through the establish mechanism and grant the waiver and forward to USD (A&T) and ASD (C3I) (DoD CIO) for concurrence. Lastly, all waivers of the standards contained in the Modeling and Simulation Domain Annex must be submitted through the M&S management office of the responsible DoD Component to the Defense Modeling and Simulation Office (DMSO). DMSO will then coordinate and administratively process a recommended disposition to the Executive Council for Modeling and Simulation (EXCIMS). EXCIMS will submit their recommendation to the USD (A&T) for approval with the concurrence of the DoD CIO.

Each DoD Component and cognizant OSD authority is requested to provide a new or revised implementation plan to the USD (A&T) and the ASD (C3I) (DoD CIO). Revision of an existing plan is due within 60 days while a new plan is due within 90 days from the date of this memorandum. These plans must include consideration of JTA implementation for existing capabilities meeting the criteria provided in the second paragraph.

The JTA is a living document and will continue to evolve with the technologies, marketplace, and associated standards upon which it is based. The JTA is the minimum set of performance based primarily non-governmental standards needed to maximize interoperability and affordability within DoD and hence is entirely consistent with Acquisition Reform principles and practices. Tests and exercises will be used to evaluate the JTA implementation progress.

Addressees are requested to assure the widest distribution of this memorandum. Request Director, Joint Staff forward this memorandum to Unified Combatant Commands.

---signed---	---signed---	---signed---
Jacues S. Gansler	Arthur L. Money	DOUGLAS D. BUCHHOLZ
Under Secretary of Defense	Senior Civilian Official	Lieutenant General,
USA		
(Acquisition and Technology)		Director for C4 Systems
		The Joint Staff

MEMORANDUM FROM
THE ASSISTANT SECRETARY OF DEFENSE

March 30, 1995

MEMORANDUM FOR UNDER SECRETARIES OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY (RD and A)
ASSISTANT SECRETARY OF THE NAVY (RD and A)
ASSISTANT SECRETARY OF THE AIR FORCE
(ACQUISITION) (SAF/AQ)
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM), Version 2.0

My memorandum dated June 23, 1994 established the TAFIM as the single framework to promote the integration of Department of Defense (DoD) information systems, expanding the opportunities for interoperability and enhancing our capability to manage information resources across the Department. The latest version of the TAFIM, Version 2.0, is complete and fully coordinated. Version 2.0 consists of seven volumes as shown in the attachment. The TAFIM will continue to guide and enhance the evolution of the Department's information systems technical architectures.

I want to reiterate two important points that I made in my June 1994 memorandum. First, the Department remains committed to a long range goal of an open systems environment where interoperability and cross functional integration of our systems and portability/reusability of our software are key benefits. Second, the further selection and evaluation of migration systems should take into account this long range goal by striving for conformance to the TAFIM to the extent possible.

Effectively immediately, new DoD information systems development and modernization programs will conform to the TAFIM. Evolutionary changes to migration systems will be governed by conformance to the TAFIM.

The TAFIM is maintained by the Defense Information Systems Agency (DISA) and is available electronically via the DISA On-Line Standards Library. Hardcopy is available through the Defense Technical Information Center. The TAFIM is an evolving set of documents and comments for improving may be provided to DISA at any time. The DISA action officer is Mr. Bobby Zoll, (703) 735-3552. The OSD action officer is Mr. Terry Hagle, (703) 604-1486.

s/Emmett Paige, Jr.

OFFICE OF THE ASSISTANT SECRETARY OF DEFENSE
3300 DEFENSE PENTAGON
WASHINGTON, DC 20301-3300

C3I January 2, 1997

MEMORANDUM FOR UNDER SECRETARIES OF DEFENSE
ASSISTANT SECRETARY OF THE ARMY
ASSISTANT SECRETARY OF THE NAVY
ASSISTANT SECRETARY OF THE AIR FORCE
(ACQUISITION)(SAF/AQ)
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Technical Architecture Framework for Information Management (TAFIM), Version 3.0

As established by my memorandum dated March 30, 1995, the TAFIM is the Department's technical architecture framework to promote the integration of Department of Defense (DoD) information systems, expanding the opportunities for interoperability and enhancing our capability to manage information resources across the Department. The latest version of the TAFIM, Version 3.0, is a fully coordinated maintenance update to Version 2.0. It consists of eight volumes as shown in attachment 1. The substantive changes to Volumes 2, 5, and 7 together with the Joint Technical Architecture (JTA) and the C4ISR Integration Task Force recommendations, will provide a firm foundation to continue refinement of the Department's technical architecture strategy.

Though the TAFIM will continue to guide and enhance the evolution of the Department's information systems technical architectures, the mandate for the use of the JTA as established in the August 22, 1996, memorandum remains in effect. For applicable systems, the JTA's specific guidance replaces the general standards guidance described in the TAFIM. The TAFIM establishes our long range goal of an open systems environment and provides the general direction. The JTA describes our specific path. I strongly encourage the further selection and evaluation of migration systems based on their potential for compliance with the TAFIM/JTA to the maximum extent possible.

The TAFIM is maintained by the Defense Information Systems Agency (DISA) and is available electronically via the DISA On-Line Standards Library. The TAFIM is an evolving set of documents, and comments for its improvement may be provided to DISA at any time. The DISA action officers are Ms. Virginia Conway, (703) 735-3552 and Mr. John Mitchell, (703) 607-6289. My point of contact for this action is Mr. Terry Hagle, who is assigned to the office of the Deputy Assistant Secretary of Defense for Command, Control and Communications, telephone number (703) 604-1486 or Mr. Samuel J. Worthington, (703) 604-1584.

-----signed ----
Emmett Paige, Jr.

Attachment

OFFICE OF THE SECRETARY OF DEFENSE
1000 Defense Pentagon
Washington, DC 20301-1000

February 23, 1998

MEMORANDUM FOR: SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS
UNDER SECRETARIES OF DEFENSE
DIRECTOR, DEFENSE RESEARCH AND ENGINEERING
ASSISTANT SECRETARIES OF DEFENSE
GENERAL COUNSEL OF THE DEPARTMENT OF DEFENSE
INSPECTOR GENERAL OF THE DEPARTMENT OF DEFENSE
DIRECTOR, OPERATIONAL TEST AND EVALUATION
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTOR OF ADMINISTRATION AND MANAGEMENT
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF

SUBJECT: Strategic Direction for a DoD Architecture Framework

The Defense Science Board concluded that a key means for ensuring interoperable and cost-effective military systems is to establish comprehensive architectural guidance for all of DoD. The Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework Version 1.0 was subsequently offered to all DoD Components for use in June of 1996. Many of you began to use it was positive results in developing your C4ISR architecture.

Representatives from the Unified Commands, Services, and Agencies, working under the auspices of the DoD Architecture Coordination Council (ACC), incorporated lessons learned from the trial implementation of Version 1.0 of the Framework and collaboratively produced Version 2.0 that more completely satisfies the need for comprehensive DoD-wide architectural guidance as recommended by the Defense Science Board. Further evolution of the Framework will be based upon experience gained and the collective needs of the Department.

The utilization of C4ISR Architecture Framework Version 2.0 will allow architectures to be compared and integrated within DoD Components and across joint boundaries so that Warfighter interoperability and C4ISR investment decisions can be addressed from a common frame of reference. Experiences with Version 1.0 demonstrate that the concepts and methodology embodied in the C4ISR Architecture Framework can be applied across the DoD community. Further, Version 2.0 C4ISR Architecture Framework is wholly consistent with the DoD Chief

Information Officer’s (CIO) responsibility to develop and implement an agency-wide architecture “model” and an Information Technology Architecture (ITA) which conforms to this model.

We see the C4ISR Architecture Framework as a critical element of the strategic direction in the Department, and accordingly direct that all on-going and, planned C4ISR or related architectures be developed in accordance with Version 2.0. Existing C4ISR architectures will be redescribed in accordance with the Framework during appropriate revision cycles. We also direct all addressees examine the C4ISR Architecture Framework as a basis for a single architecture framework for all functional areas/domains within Department. The ACC with the active participation of the C4ISR community has established a process by which the evolution of the C4ISR Architecture Framework will continue in a manner responsive to the needs of operations and acquisition elements of the DoD.

The successful evolution to a DoD Architecture Framework is dependent upon your active participation and a continued commitment to interoperable and cost effective military systems.

Request the Director, Joint Staff forward this memorandum to the Unified Commands.

Version 2.0 of the C4ISR Architecture Framework is available at www.cisa.osd.mil.

--signed--
Jacques S. Gansler
Under Secretary of Defense
(Acquisition and Technology)

--signed--
Anthony M. Valletta
Acting Assistant Secretary
of Defense (C3I)

--signed--
DOUGLAS D. BUCHHOLZ
Lieutenant General, USA
Director for C4 Systems
The Joint Staff

APPENDIX C - ABBREVIATIONS, ACRONYMS, AND GLOSSARY

C.1 ABBREVIATIONS/ACRONYMS

A and T	Acquisition and Technology
AIS	Automated Information System
AITS	Adopted Information Technology Standards
ANSI	American National Standards Institute
API	Application Program Interface
ASD	Assistant Secretary of Defense
C2	Command and Control
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CASE	Computer-Assisted Software Engineering
CCB	Configuration Control Board
CIM	Corporate Information Management
CISS	Center for Information System Security
CM	Configuration Management
COTS	commercial-off-the-shelf
DBMS	Database Management System
DDDS	Defense Data Dictionary System
DGSA	DoD Goal Security Architecture
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DISN	Defense Information System Network
DITSCAP	Defense Information Technology Security Certification Accreditation Procedures
DoD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
DSSP	Defense Standardization and Specification Program
DT&E	Developmental Test and Evaluation
EEI	External Environment Interface

FEA	Functional Economic Analyses
FIPS	Federal Information Processing Standard
FPI	Functional Process Improvement
GOA	Generic Open Architecture
GOTS	Government-off-the-shelf
HCI	Human-Computer Interface
HDBK	Handbook
I-CASE	Integrated Computer-Assisted Software Engineering
ICD	Interface Control Document
IDEF	ICAM Definition Method for Integrated Computer System Manufacturing
IEEE	Institute of Electrical and Electronic Engineers
ILS	Integrated Logistics Support
IM	Information Management
IPT	Integrated Product Team
ISO	International Organization for Standardization
IT	Information Technology
ITSG	Information Technology Standards Guidance
JTA	Joint Technical Architecture
LCM	Life-Cycle Management
MAISRC	Major Automated Information System Review Council
METL	Mission Essential Task List
MNS	Mission Need Statement
NDI	Non-Developmental Item
NGCR	Next-Generation Computer Resources
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OMG	Object Management Group
OSA	Open System Architecture
OSD	Office of the Secretary of Defense
OSE	Open Systems Environment
OT&E	Operational Test and Evaluation
PC	Personal Computer

PDSS	Post-Deployment Software Support
PMO	Program Management Office
POM	Program Objective Milestones
POSIT	Profiles for Open System Internetworking Technologies
POSIX	Portable Operating System Interface
QA	Quality Assurance
RFP	Request for Proposal
RMP	Risk Management Plan
SAE	Society of Automotive Engineers
SBA	Standards-Based Architecture
SDM	System Decision Memorandum
SPE	Software Performance Engineering
STD	Standard
T&E	Test and Evaluation
TAFIM	Technical Architecture Framework for Information Management
TEMP	Test and Evaluation Master Plan
TRM	Technical Reference Model
TSR	Trade Study Report
UJTL	Unified Joint Task List

C.2 GLOSSARY[1]

Application – The use of capabilities [services and facilities] TAFIM provided by an information system specific to the satisfaction of a set of user requirements. [TAFIM Version 3.0, Volumes 1 and 3]

Application Area Profile – A profile created from multiple standards that specify multiple, diverse types of functionality for a particular application area (e.g., database, networking, graphics, operating system). (IEEE) [JTA Version 2.0, 26 May 1998]

Application Platform:

- The collection of hardware and software components that provide the services used by support and mission-specific software applications. [TAFIM Version 3.0, Volumes 1 and 3]
- A set of resources, including hardware and software, that support the services on which application software will execute. The application platform provides services at its interfaces that, as much as possible, make the specific characteristics of the platform transparent to the application software. [TAFIM Version 3.0, Volumes 1 and 3] [JTA Version 2.0, 26 May 1998]

Application Portability Profile (APP) **–** The structure that integrates Federal, national, international, and other specifications to provide the functionality necessary to accommodate the broad range of Federal information technology requirements. [TAFIM Version 3.0, Volumes 1 and 3]

Application Environment Profile (AEP) **–** "A profile, specifying a completed and coherent specification of the Open System Environment (OSE), in which the standards, options, and parameters chosen are necessary to support a class of applications." (IEEE) [JTA Version 2.0, 26 May 1998]

Application Program Interface (API):

- The interface, or set of functions, between the application software and the application platform. (IEEE) [TAFIM Version 3.0, Volumes 1 and 3] (NIST Special Publication 500-230; TAFIM, Version 3.0, Volumes 1 and 3)

- The means by which an application designer enters and retrieves information. [TAFIM Version 3.0, Volumes 1 and 3]

[1] Listing of Reference Sources:
Where a definition does not contain a specific reference, the de facto reference is the JTA (Joint Technical Architecture).
C4ISR Architecture Framework, Version 2.0, 18 December 1997.
Joint Technical Architecture, Version 3.0, 15 Nov 1999.
IEEE (Draft Guide for Developing User Organization Open Systems Environment (OSE) Profiles), P1003.23/D0.8, April 1997.

- A specification of function-call conventions that defines an interface to a service. If two incompatible computers both support the same API, then a single version of source code should compile on each.

Application Software – “Software that is specific to an application and is composed of programs, data, and documentation.” (IEEE)

Application Software Entity – Mission-area and support applications. A common set of support applications forms the basis for the development of mission-area applications. Mission-area should be designed and developed to access this set of common support applications. Applications access the Application Platform via a standard set of APIs. (TAFIM) [JTA Version 2.0, 26 May 1998]

Architecture:

- The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time. – C4ISR Architecture Framework (Ver. 1.0) [IEEE] [C4ISR]

- Architecture has various meanings, depending upon its contextual usage. (1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE]TAFIM] (2) Organizational structure of a system or component. [IEEE] [TAFIM, Version 3.0, Volumes 1 and 3]

- An architecture is a composition of (1) components (including humans) with their functionality defined (Technical), (2) requirements that have been configured to achieve a prescribed purpose or mission (Operational), and (3) their connectivity with the information flow defined (System). [JTA Version 2.0, 26 May 1998]

- Architecture has various meanings depending upon its contextual usage. (1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE] (2) Organizational structure of a system or component. [IEEE] [TAFIM, Version 3.0, Volumes 1 and 3]

- Architecture: Baseline and Target-Defined and are significant parts of the technical management planning information (previously the technical management plan [TMP]). [DoD 8020.1-M with Change 1] [TAFIM, Version 3.0, Volumes 1 and 3]

Architecture Description – A representation, as of a current or future point in time, of a defined “domain” in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function. [C4ISR]

Architecture-Description Products – Graphical, textual, and tabular items that are developed in the course of building a given architecture description and that describe characteristics pertinent to its purpose. When completed, this set of products constitutes the architecture description. [C4ISR]

Architecture, Infrastructure – Identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities including hardware builds versus schedule and costs. [TAFIM, Version 3.0, Volumes 1 and 3]

Architectural Structure – Provides the conceptual foundation of the basic architectural design concepts, the layers of the technical architecture, the services provided at each layer, the relationships between the layers, and the rules for how the layers are interconnected. [TAFIM, Version 3.0, Volumes 1 and 3]

Availability – The probability that system functional capabilities are ready for use by a user at any time, where all time is considered, including operations, repair, administration, and logistic time. Availability is further defined by system category for both routine and priority operations. [TAFIM, Version 3.0, Volumes 1 and 3]

Baseline – A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and that can be changed only through formal change control procedures or a type of procedure such as configuration management. [IEEE] [TAFIM, Version 3.0, Volumes 1 and 3]

Commercial Item:

- Any item customarily used by the general public for other than governmental purposes that has been sold, leased, or licensed to the general public, or that has been offered for sale, lease, or license to the general public.

- Any item that evolved from an item described in 1) above through advances in technology or performance that is not yet available in the commercial market, but will be available in time to meet the delivery requirements of the solicitation.

- Any item that, but for modifications of a type customarily available in the commercial market or minor modifications made to meet DoD requirements, would satisfy the criteria in 1) or 2) above.

- Any combination of items meeting the requirements of 1, 2, or 3 above or 5 below that are of a type customarily combined and sold in combination to the general public.

- Installation services, maintenance services, repair services, training services, and other services if such services are procured for support of any item referred to paragraphs 1, 2, 3, or 4 above, if the sources of such services offer such services to the general public and the DoD simultaneously and under similar terms and conditions and offers to use the same work force for providing DoD with such services as the source used for providing such services to the general public.

- Services offered and sold competitively, in substantial quantities, in the commercial marketplace based on established catalog prices of specific tasks performed and under standard commercial terms and conditions.

- Any item, combination of items, or service referred to in 1 through 6 above notwithstanding the fact that the item or service is transferred between or among separate divisions, subsidiaries, or affiliates of a contractor.

- A non-developmental item developed exclusively at private expense and sold in substantial quantities, on a competitive basis, to State and local governments.

(DRAFT 6/30/95 NDI HANDBOOK/ Federal Acquisition Streamlining Act of 1994 DoD 5000.37H) [JTA Version 2.0, 26 May 1998]

Commercial-Off-the-Shelf (COTS) – See the definition of Commercial Item found above. (OS-JTF 1995) Refers to an item of hardware or software that has been produced by a contractor and is available for general purchase. Such items are at the unit level or higher. Such items must have been sold and delivered to Government or commercial customers, must have passed customer's acceptance testing, be operating under customer's control, and within the user environment. Further, such items must have meaningful reliability, maintainability, and logistics historical data. [TAFIM] [JTA Version 2.0, 26 May 1998]

Communications Medium – A means of data transmission. [C4ISR Version 2.0]

Communications Network – A set of products, concepts, and services, that enable the connection of computer systems for the purpose of transmitting data and other forms (e.g., voice and video) between the systems. [TAFIM, Version 3.0, Volumes 1 and 3]

Communications Services – A service of the Support Application entity of the Technical Reference Model (TRM) that provides the capability to compose, edit, send, receive, forward, and manage electronic and voice messages and real-time information exchange services in support of interpersonal conferencing. [TAFIM, Version 3.0, Volumes 1 and 3]

Communications System – A set of assets (transmission media, switching nodes, interfaces, and control devices) that will establish linkage between users and devices. [TAFIM, Version 3.0, Volumes 1 and 3]

Configuration Management:

- A discipline applying technical and administrative direction and surveillance to: (1) identify and document the functional and physical characteristics of a configuration item, (2) control changes to those characteristics, and (3) record and report changes to processing and implementation status. [JTA Version 2.0, 26 May 1998]

- A discipline applying technical and administrative direction and surveillance to: (a) identify and document the functional and physical characteristics of a configuration item, (b) control changes to those characteristics and, (c) record and report changes to processing and implementation status. [TAFIM, Version 3.0, Volumes 1 and 3]

Connectivity Service – A service area of the External Environment entity of the Technical Reference Model that provides end-to-end connectivity for communications through three transport levels (global, regional, and local). It provides general and applications-specific service to platform end devices. [TAFIM, Version 3.0, Volumes 1 and 3]

Core Services – JTA Core Services are common service areas, interfaces, and standards applicable to all DoD systems to support interoperability [JTA Version 2.0].

Data – A representation of individual facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means [IEEE] [C4ISR Version 2.0]

Data Element:

- A basic unit of information having a meaning and that may have sub-categories (data items) of distinct units and values. [TAFIM, Version 3.0, Volumes 1 and 3]

- A basic unit of data having a meaning and distinct units and values. (Derived from 8320.1) A uniquely named and defined component of a data definition; a data "cell" into which data items (actual values) can be placed; the lowest level of physical representation of data. [C4ISR Version 2.0]

Data Entity – The representation of a set of people, objects, places, events, or ideas that share the same characteristic relationships. [C4ISR Version 2.0]

Data Integrity – The state that exists when computerized data is the same as that in the source documents and has not been exposed to accidental or malicious alteration or destruction. The property that data has not been exposed to accidental or malicious alteration or destruction. [JTA Version 2.0, 26 May 1998]

Data Interchange Service – A service of the Platform Entity of the Technical Reference Model that provides specialized support for the interchange of data between applications on the same or different platforms. [TAFIM, Version 3.0, Volumes 1 and 3]

Data Management Service – A service of the Platform Entity of the Technical Reference Model that provides support for the management, storage, access, and manipulation of data in a database. [TAFIM, Version 3.0, Volumes 1 and 3]

Database Utility Service – A Service of the Support Application Entity of the Technical Reference Model that provides the capability to retrieve, organize, and manipulate data extracted from a database. [TAFIM, Version 3.0, Volumes 1 and 3]

Direct Interface – Direct Interface is an interface that defines a service/consumer relationship between adjacent entity/sub-entity layers in the reference model. Direct Interface is the connection between an entity sending (or receiving) data with another entity receiving (or sending) data for transmission of that data along the routing path. [SAE AS4893]

Distributed Database:

- A database not stored in a central location but dispersed over a network of interconnected computers.

- A database under the overall control of a central database management system but whose storage devices are not all attached to the same processor. (3) A database physically located in two or more distinct locations. [TAFIM, Version 3.0, Volumes 1 and 3]

Domain:

- A domain represents a grouping of systems sharing common functional, behavioral, and operational requirements. [JTA Version 2.0].

- Elements at any level, from DoD as a whole down to individual functional areas or groups of functional areas. [C4ISR Version 2.0]

- A distinct functional area that can be supported by a family of systems with similar requirements and capabilities. An area of common operational and functional requirements. [JTA Version 2.0, 26 May 1998]

Entity:

- An active element within an open-system layer (e.g., session entity, transport entity). It can represent one layer, or several layers of the OSI Reference Model. One layer can include several entities. [TAFIM, V3, Volume 4]

- Entities are the elements of the model that contain the services and interfaces definitions subsequently used to select and refine a set of standards. Entities contain the major service areas which may be common across several entities.

Environment:

- A framework that provides a set of services and consists of the following: an integration mechanism, a set of resources (i.e., software, tools), a methodology and a set of standards.

- In the context of the COE, all software running from the time the computer is rebooted to the time the system is ready to respond to operator queries after operator login. This software includes the operating system, security software, installation software, windowing environment, COE services, etc. The environment is subdivided into a runtime environment and a software development environment. [DII COE, Version 3.0]

External Environment Interface (EEI) – The interface that supports information transfer between the application platform and the external environment. [JTA Version 2.0, 26 May 1998]

Function – Appropriate or assigned duties, responsibilities, missions, tasks, powers, or duties of an individual, office, or organization. A functional area is generally the responsibility of a PSA (e.g., personnel) and can be composed of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviews). [TAFIM, Version 3.0, Volumes 1 and 3]

Functional Area – A major area of related activity, such as Ballistic Missile Defense, Logistics, or C2 support. [C4ISR Version 2.0]

Functional Architecture – The framework for developing applications and defining their interrelationships in support of an organization's information architecture. It identifies the major functions or processes an organization performs and their operational interrelationships. [TAFIM, Version 3.0, Volumes 1 and 3]

Functional Reference Model (FRM) – A Representation of a system(s) within an operational domain that defines the functions required for a system(s) to perform and the information flow associated with those functions. There may exist more than one FRM within a domain.

Generic Level Of Interoperability – For each system, the highest level within the capabilities matrix at which that system implements all the criteria for that LISI level. [C4ISR Version 2.0]

Hardware:

1. Physical equipment, as opposed to programs, procedures, rules, and associated documentation.
2. Contrast with software. [TAFIM, Version 3.0, Volumes 1 and 3]

Information:

1. Any communication or representation of knowledge such as facts, data, or opinions, in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual forms. [TAFIM, Version 3.0, Volumes 1 and 3]
2. The refinement of data through known conventions and context for purposes of imparting knowledge. [C4ISR Version 2.0]

Information Domain – A set of commonly and unambiguously labeled information objects with a common security policy that defines the protections to be afforded the objects by authorized users and information management systems. [TAFIM, Version 3.0, Volumes 1 and 3]

Information-Exchange Requirement – A requirement for the content of an information flow. Associated with an IER are such performance attributes as information size, throughput, timeliness, quality, and quantity values. [C4ISR Version 2.0]

Information Management (IM) **–** The creation, use, sharing, and disposition of information as resource critical to the effective and efficient operation of functional activities. The structuring of functional processes to produce and control the use of data and information within functional activities, information systems, and computing and communications infrastructures. [TAFIM, Version 3.0, Volumes 1 and 3]

Information Resources Management (IRM) **–** The planning, budgeting, organizing, directing, training, promoting, controlling, and management activities associated with the burden (cost), collection, creation, use, and dissemination of information by Agencies and includes managing information and related resources, such as Federal information-processing (FIP) resources. [TAFIM, Version 3.0, Volumes 1 and 3]

Information Processing:

1. Profiles are defined in this paper as a hierarchical tree-structured collection of services needed to support applications that address a specific mission area. [IEEE. Taxonomy Pub.]
2. An activity that conveys collated data, entities, or elements in a systematic routine that addresses a specific area.
3. Provide the data formats and instruction-processing specifications required to represent and manipulate data to meet information-technology (IT) mission needs. [JTA version 2.0, 26 May 1998]
4. The systematic performance of operations upon data such as handling, merging, sorting, and computing. Note: The semantic content of the original data should not be changed. The semantic content of the processed data may be changed. [Federal Standard 1037c]

Information Technology (IT) – The term "information technology," with respect to an executive agency means any equipment or interconnected system or subsystem of equipment used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the executive agency. For purposes of the preceding sentence, equipment is used by an executive agency if the equipment is used by the executive agency directly or is used by a contractor under a contract with the executive agency that requires the use of such equipment, or (ii) requires the use, to a significant extent, of such equipment in the performance of a service or the furnishing of a product. Information systems also include training (e.g., tutorials and online help), support tools (e.g., programs for software development, self-test diagnostics), and system management aids (e.g., system administration).

The term "information technology" includes computers, ancillary equipment, software, firmware and similar procedures, services (including support services), and related resources. The term "information technology" does not include any equipment that is acquired by a Federal contractor incidental to a Federal contract. (Information Technology Management Reform Act of 1996. See http://www.dtic.mil/c3i/cio/itmra.Annot.html [JTA Version 2.0, 26 May 1998]

Infrastructure – Infrastructure is used with different contextual meanings. Infrastructure most generally relates to, and has, a hardware orientation but it is frequently more comprehensive and includes software and communications. Collectively, the structure must note that just citing standards for designing an architecture or infrastructure does not include functional and mission-area requirements for performance. Performance requirement metrics must be an inherent part of an overall infrastructure to provide performant interoperability and compatibility. It identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. [TAFIM, Version 3.0, Volumes 1 and 3]

Integration:

- Integration is the result of an effort that joins two or more similar products such as individual system elements, components, modules, processes, databases, or other entities, and produces a new product that functions, as a replacement for the two or more similar but less capable entities (products), in a framework or architecture in a seamless manner. Institute of Electrical and Electronic Engineers (IEEE) Standard (STD) 610.12 defines an “integration architecture” as a framework for combining software components, hardware components, or both to an overall system. [IEEE] [TAFIM, Version 3.0, Volumes 1 and 3]

- Two or more software applications that must run on the same physical processor(s) and under the same operating system. [JTA Version 2.0, 26 May 1998]

Interface (I/F):

- In a system, a shared boundary, i.e., the boundary between two subsystems or two devices. (188)

- A shared boundary between two functional units, defined by specific attributes, such as functional characteristics, common physical interconnection characteristics, and signal characteristics.

- A point of communication between two or more processes, persons, or other physical entities.

- A point of interconnection between user terminal equipment and commercial communications facilities.

- To interconnect two or more entities at a common point or shared boundary. [FED-STD 1037c]

Interface View

The view of the DoD TRM that contains all of the interfaces (e.g., 4L, 4D, 3D, 3L, 2D, 2L, etc.) Details of services are not contained in this view of the model.

Interoperability

- The ability of two or more systems or components to exchange and use information. [IEEE 610.12]. (2) The ability of the systems, units, or forces to provide and receive services from other systems, units, or forces, and to use the services so interchanged to enable them to operate effectively together. The conditions achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. [TAFIM, Version 3.0, Volumes 1 and 3]

- The ability of two or more systems or components to exchange data and use information. [IEEE]

- The ability of two or more systems to exchange information and to mutually use the information that has been exchanged. (Army Science Board) [JTA Version 2.0, 26 May 1998]

Layer – The term is used in conjunction with the interface view (originally derived from the SAE GOA model) to identify specific interfaces between model elements. The reader is referred to the SAE AS 4893 Standard for more information on the subject. The term layer in the services view of the model is used as a generic.

Legacy Environments – Legacy environments could be called legacy architectures or infrastructures and, as a minimum, consist of a hardware platform and an operating system. Legacy environments are identified for phase-out, upgrade, or replacement. All data and applications software that operates in a legacy environment must be categorized for phase-out, upgrade, or replacement. [TAFIM, Version 3.0, Volumes 1 and 3]

Legacy Systems – Systems that are candidates for phase-out, upgrade, or replacement. Generally, legacy systems are in this category because they do not comply with data standards or other standards. Legacy system workloads must be converted, transitioned, or phased out (eliminated). Such systems may or may not operate in a legacy environment. [TAFIM, Version 3.0, Volumes 1 and 3]

Logical Interface – Logical Interface is an interface that defines a peer-to-peer relationship between entities within the same entity layer of the reference model. Logical Interface is the requirement associated with establishing a data interchange between a source of data and the end user of the data. The end user of the data must be identified including the requirements for the data, and the source supplying the data must also be identified. Data routing is transparent to logical interface entities. Routing of the data should not be a concern to the source and end user because the routing (i.e., direct requirements) is transparent to the entities. [SAE AS4893]

Migration Systems – An existing AIS, or a planned and approved AIS, that has been officially designated to support common processes for a functional activity applicable to use DoD-wide or DoD Component-wide. Systems in this category, even though fully deployed and operational, have been determined to accommodate a continuing and foreseeable future requirement and, consequently, have been identified for transitioning to a new environment or infrastructure. A migration system may need to undergo transition to the standard technical environment and standard data definitions being established through the Defense IM Program, and must "migrate" toward that standard. In that process it must become compliant with the Reference Model and the Standards Profile. A system in this category may require detailed analysis that involves a total redesign, reprogramming, testing, and implementation because of a new environment and how the "users" have changed their work methods and processes. The detailed analysis may identify the difference between the "as is" and the "to be" system. [TAFIM, Version 3.0, Volumes 1 and 3]

Mission:

- An objective. [JTA Version 2.0, 26 May 1998]

- An objective together with the purpose of the intended action.

  1. Note: Multiple tasks accomplish a mission. (SPAWAR) [C4ISR Version 2.0]

Mission Area – The general class to which an operational mission belongs. [C4ISR Version 2.0]

Model – A model is a representation of an actual or conceptual system that involves mathematics, logical expressions, or computer simulations that can be used to predict how the system might perform or survive under various conditions or in a range of hostile environments. [DSMC 8th Ed. of the Glossary]

Multimedia Service – A Service of the TRM that provides the capability to manipulate and manage information products consisting of text, graphics, images, video, and audio. [TAFIM, Version 3.0, Volumes 1 and 3]

Multimedia Object – A composite object consisting of various different types of related temporal and logical content intended for presentation to a user. [TAFIM, Version 3.0, Volumes 1 and 3]

Node:

- An element of architectures that may represent a role, an organization, a facility, or even an individual workstation, depending on the purpose and the level of detail needed in the architecture description. (As used in the Framework; no formal definition arrived at yet.)

- A primitive that is a component of a network. Again, not limited to a node in a communications network. Can be combined with arcs (using NODE-ASSOCIATION, not depicted in Figure 3-4) to represent virtually any network or graph structure. – [C4ISR Version 2.0]

Normalization Rules – Rules used to identify the movement or removal of elements (e.g., standards) across the JTA Core or annexes. [JTA, Version 2.0]

Open Specifications – Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards. [TAFIM, Version 3.0, Volumes 1 and 3]

Open System:

- A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [TAFIM, Version 3.0, Volumes 1 and 3]

- A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and to interact with users in a style that facilitates portability. An open system is characterized by the following:

  - Well-defined, widely used, non-proprietary interfaces/protocols.
  - Use of standards developed/adopted by industry-recognized standards bodies.
  - Definition of all aspects of system interfaces to facilitate new or additional systems capabilities for a wide range of applications.
  - Explicit provision for expansion or upgrading through the incorporation of additional or higher-performance elements with minimal impact on the system. [IEEE] as modified by the Tri-Service Open Systems Architecture Working Group) [JTA Version 2.0, 26 May 1998]

Open Systems Approach – An open systems approach is a business approach that emphasizes commercially supported practices, products, specifications and standards. The approach defines, documents, and maintains a system technical architecture that depicts the lowest level of system configuration control. This architecture clearly identifies all the performance characteristics of the system including those that will be accomplished with an implementation that references open standards and specifications. [JTA Version 2.0, 26 May 1998]

Open Systems Environment (OSE) – The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of applications, data, or people, as specified by information technology standards and profiles. [TAFIM, Version 3.0, Volumes 1 and 3]

Operational Architecture – Descriptions of the tasks, operational elements, and information flows required to accomplish or support a warfighting function. [C4ISR Version 2.0]

Operational Architecture View – A doctrine-driven description of the tasks, operational elements, and information flows required to accomplish or support a military operation. [C4ISR Version 2.0]

Operational Element – An organization or a portion of an organization or a type of organization. Note: Operational Architectures typically represent an operational element within an operational node. [C4ISR Version 2.0]

Operational Node – A node that performs a role or mission. [C4ISR Version 2.0]

Operating System Service – A core service of the Platform entity of the Technical Reference Model that is needed to operate and administer the application platform and provide an interface between the application software and the platform (e.g., file management, input/output, print spoolers). [TAFIM, Version 3.0, Volumes 1 and 3]

Operational Architecture (OA) –A description (often graphical) of the operational elements, assigned tasks, and information flows required to support the warfighter. It defines the type of information, the frequency of the exchange, and what tasks are supported by these information exchanges. [JTA Version 2.0, 26 May 1998]

Organization – An administrative structure with a mission. Organization is used here in a very broad sense. Includes military organizations, agencies, units, OPFACs, and even governments. [C4ISR Version 2.0]

Peer Entity – In layered systems, one of a set of entities in the same layer or the equivalent layer of another system or level abstraction. [FED-STD 1037c]

Peer-to-Peer Interface – The functional and physical characteristics required to exist at a common boundary or connection between peer entities.

Platform:

- The entity of the Technical Reference Model that provides common processing and communication services provided by a combination of hardware and software and required by users, mission-area applications, and support applications. [TAFIM, Version 3.0, Volumes 1 and 3]

- A system that is a physical structure hosting systems or systems components. Note: A kind of system element in the CADM. [C4ISR Version 2.0]

Portability:

- The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE]

- A quality metric used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software-system environment. [IEEE]

- The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TAFIM, Version 3.0, Volumes 1 and 3]

Process – A group of logically related activities required to execute a specific task or group of tasks. (Army Systems Architecture Framework). Note: Multiple activities make up a process. [C4ISR Version 2.0]

Process Model – Provides a framework for identifying, defining, and organizing the functional strategies, functional rules, and processes needed to manage and support the way an organization does or wants to do business. Provides a graphical and textual framework for organizing the data and processes into manageable groups to facilitate their shared use and control throughout the organization. [TAFIM, Version 3.0, Volumes 1 and 3]

Profile – A set of one or more base standards and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [TAFIM, Version 3.0, Volumes 1 and 3]

Requirement – A need or demand. A subtype of guidance. May be specified in other guidance or derived from necessity and circumstances. [C4ISR Version 2.0]

Scalability:

- The capability to adapt hardware or software to accommodate changing workloads. (OS-JTF)

- The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). The ability to grow to accommodate increased workloads. [JTA Version 2.0, 26 May 1998]

- The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII] The capability to grow to accommodate increased workloads. [TAFIM, Version 3.0, Volumes 1 and 3]

Seamless Interface – Ability of facilities to call one another or exchange data with one another in a direct manner. Integration of the user interface that allows a user to access one facility through another without any noticeable change in user interface conventions. [TAFIM, Version 3.0, Volumes 1 and 3]

Service – A collection of components organized to accomplish a specific function or set of functions. [IEEE] [C4ISR Version 2.0]

Service Area: – a higher level abstraction or category of services that consist of a number of lesser or subordinate grouped set of services. Also known as a “Service Category.”

Service View – The view of the DoD TRM that contains all of the services contained within the entities of the model. Details of interfaces are not contained in this view of the model other than a high level Application Programming Interface (API) and an External Environment Interface (EEI).

Software-Item – A set of instructions that govern the operation of data-processing equipment. Includes firmware, software applications, operating systems, and embedded software. [C4ISR Version 2.0]

Specific Level Of Interoperability – The highest LISI level between a pair of systems that provides common procedures, applications, infrastructure, and data in support of information-service exchange. [C4ISR Version 2.0]

Standard:

- An agreement for a procedure, product, or relationship. [C4ISR Version 2.0]
- A document that establishes uniform engineering and technical requirements for processes, procedures, practices, and methods. Standards may also establish requirements for selection, application, and design criteria of material. [JTA Version 2.0, 26 May 1998]

Standards Technology Forecast – A detailed description of emerging technology standards relevant to the systems and business processes covered by the architecture. [C4ISR Version 2.0]

Stovepipe System – A system, often dedicated or proprietary, that operates independently of other systems. The stovepipe system often has unique, nonstandard characteristics. [TAFIM, Version 3.0, Volumes 1 and 3]

System:

- A collection of components organized to accomplish a specific function or set of functions. [IEEE] [C4ISR Version 2.0]
- People, machines, and methods organized to accomplish a set of specific functions.
- An integrated composite of people, products, and processes that provides a capability or satisfies a stated need or objective. [JTA Version 2.0, 26 May 1998]

Systems Architecture (SA) – A description, including graphics, of the systems and interconnections providing for or supporting a warfighting function. The SA defines the physical connection, location, and identification of the key nodes, circuits, networks, warfighting platforms, etc., and allocates system and component performance parameters. It is constructed to satisfy Operational Architecture requirements in the standards defined in the Technical Architecture. The SA shows how multiple systems within a domain or an operational scenario link and interoperate and may describe the internal construction or operations of particular systems in the SA. [JTA Version 2.0, 26 May 1998]. [C4ISR Version 2.0]

Systems Architecture View – A description, including graphics, of systems and interconnections providing for, or supporting, military operations; includes the physical connection, location, and identification of key nodes, circuits, networks, warfighting platforms, and specifies system and component performance parameters. [C4ISR Version 2.0]

System Element – Subset of a system that maintains a separate identity and performs a specific function. [C4ISR Version 2.0]

System Function – A data transform that supports the automation of activities or exchange requirements. [C4ISR Version 2.0]

System Migration – Process of incrementally creating a more streamlined, efficient, smaller, and cheaper suite of system(s). [C4ISR Version 2.0]

Systems Node – A node with the identification and allocation of resources (e.g., people, platforms, facilities, or systems) required to implement specific roles and missions. [C4ISR Version 2.0]

System Management Service – A service of the platform entity of the TRM that provides for the administration of the overall information system. These services include the management of information, processors, networks, configurations, accounting, and performance. [TAFIM, Version 3.0, Volumes 1 and 3]

System Technology – A detailed description of emerging technologies and specific hardware and software products. [C4ISR Version 2.0]

Tailoring – The process of using a model and adjusting it to fit specific domain or implementation requirements.

Technical Architecture (TA) – The minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements whose purpose is to ensure that a conformant system satisfies a specified set of requirements. The technical architecture identifies the services, interfaces, standards, and their relationships. It provides the technical guidelines for implementation of systems upon which engineering specifications are based, common building blocks are built, and product lines are developed. [JTA Version 2.0, 26 May 1998]

Technical Architecture View – A description of the minimal set of rules governing the arrangement, interaction, and interdependence of system architecture components; includes the technical criteria governing system services, interfaces, and relationships. [C4ISR Version 2.0]

Technical Reference Model (TRM) (Original definition considered inaccurate):

- A target framework and profile of standards for the DoD computing and communications infrastructure. [JTA Version 2.0, 26 May 1998]

- The document that identifies a target framework and profile of standards for the DoD computing and communications infrastructure. [TAFIM, Version 3.0, Volumes 1 and 3]

New Definition Proposed (DoD TRM Version 1.0 Draft):

A conceptual framework that provides the following:

- A consistent set of service and interface categories and relationships used to address interoperability and open-system issues

- Conceptual entities that establish a common vocabulary to better describe, compare, and contrast systems and components.

- A basis (an aid) for the identification, comparison, and selection of existing and emerging standards and their relationships.

The TRM framework is not an architecture, is not a set of, and does not contain, standards.

User:

- Any person, organization, or functional unit that uses the services of an information-processing system.

- In a conceptual schema language, any person or any thing that may issue or receive Commands and messages to or from the information system. [TAFIM, Version 3.0, Volumes 1 and 3]

User Interface Service – A service of the Platform entity of the Technical Reference Model that supports direct human-machine interaction by controlling the environment in which users interact with applications. [TAFIM, Version 3.0, Volumes 1 and 3]

View – Perspectives (operational, systems, and/or technical) that logically combine to describe an architecture; a view implies what architecture characteristics are to be considered and/or displayed. [C4ISR Version 2.0]

APPENDIX D - SERVICES SUMMARY

:

D.1 INTRODUCTION

The summary of services provided in this appendix will assist the reader in quickly locating a particular service. The services are listed in alphabetic order as they are found within a respective entity. For additional information on the services the reader should refer to the appropriate sections in this document. Support Services are first listed, followed by Application Services, and then External Services.

Section D.1 lists the top two levels of service detail. Section D.2 lists services down to a third level. The list is not all-inclusive, and it must be recognized that additional services can be added as the model evolves or as user requirements dictate.

D.1.1 Services Summary Top Level (Major Service Area)

Support Application Services (4.4.1.2)

Multimedia (4.4.1.2.1):

- Audio Processing
- Document Processing
- Electronic Publishing
- Image Processing
- Map Graphics
- Multimedia Processing
- Video Processing
- Text Processing

Communications (4.4.1.2.2):

- Broadcast
- Communications Conferencing
  - Enhanced Telephony
  - Organizational Messaging
  - Personal Messaging
  - Shared-Screen Teleconferencing
  - Video Teleconferencing

Business Processing (4.4.1.2.3):

- Calendar
- Calculation
- Project Management
- Spreadsheet

Environment Management (4.4.1.2.4):

- Batch Processing
- Computer-Based Training
- Information Presentation and Distribution
- Transaction Processing

Database Utilities (4.4.1.2.5):

- Networking/Concurrent Access Services
- Query Processing
- Report Generation
- Screen Generation

Engineering Support (4.4.1.2.6):

- Computer-Aided Design
- Decision Support
- Expert System
- Modeling and Simulation

D.1.2 Application Platform Services (4.4.2)

Software Engineering Services (4.4.2.1.1):

- Bindings
- Computer-Aided Software Engineering tools and environment
- Language
- Software Life Cycle Processes

User Interface Services (4.4.2.1.2):

- Character-Based User Interface
- Graphical Client-Server
- Object Definition and Management
- User Interface
- Window Management

Data-Management Services (4.4.2.1.3):

- Data Dictionary/Directory
- Database Management System
- Transaction Processing

Data-Interchange Services (4.4.2.1.4):

- Characters and Symbols
- Compression
- DoD Applications
- Document Interchange
- Hardware Applications
- Mapping
- Optical Digital Technologies
- Product Data Interchange
- Raster/Image Data Interchange
- Technical Data Interchange

Graphics Services (4.4.2.1.5):

- Device Interfaces
- Raster Graphics
- Vector Graphics

Communications Services (4.4.2.1.6):

- Application-Oriented

- Network Technologies
- Transport-Oriented

Security Services (4.4.2.1.7):

- Access Control
- Architectures and Applications
- Authentication
- Availability
- Confidentiality
- Integrity
- Non-Repudiation
- Security Labeling
- System Management

System Management Services (4.4.2.1.8):

- Configuration Control
- Fault Monitoring
- Information System Security Management
- Other Management
- Performance Monitoring
- State Management
- Usage Monitoring and Cost Allocation
- User/Group Management

Distributed-Computing Services (4.4.2.1.9):

- Client-Server
- Object
- Remote Access

Internationalization Services (4.4.2.1.10):

- Character Sets and Data Representation
- Cultural Convention
- Native Language Support

Operating System Services (4.4.2.2):

- Clock/Calendar

- Fault Management
- Kernel Operations
- Media Handling
- Operating System Object
- Real-Time Extension
- Shell and Utilities

D.1.3 External Environment (Entities with which Application Platform exchanges information) (4.4.3):

- Human Users (People)
- Information Interchange (e.g., removable memory)
- Communications (e.g., telephones, networks, cabling, packet-switching equipment.)
- Doctrinal Mechanisms (physical, administrative, personnel)

D.2 SERVICES SUMMARY DETAILED (Expanded Definitions):

D.2.1 Support Application Services

Business Processing: [Common office functions used in day-to-day operations]

Calendar	[Manage personal tasks and time, coordinate schedules]
Calculation	[routine and complex arithmetic calculations]
Project Management	[Tools that support planning, administration, and management of projects]
Spreadsheet	[Capability to create, manipulate, present information in various forms]

Communications: [Capability to send, receive, forward, manage electronic and voice messages].

Broadcast	[one-way audio, audio/video communications services]
Computer Conferencing	[conference via workstations, document exchanges
Enhanced Telephony	[Call forward, call waiting, programmed directories, teleconferencing, voice mail]
Organizational Messaging	[Send, receive, forward, display, retrieve, validate, disseminate, prioritize, manage, authenticate]
Personal Messaging	[Send, receive, forward, store, display, manage personal messages, et al.]
Shared-Screen Teleconference	[For two or more users using shared workstation windows]
Video Teleconferencing	[Two way video, full motion display]

Database Utilities: [Capabilities to retrieve, organize, and manipulate data from DBMSs]

Networking/Concurrent Access Services [Manage concurrent user access to DBMS]

Query Processing [Interactive selection, extraction, and formatting of information]

Report Generation [To define and generate hard-copy reports from DBs]

Screen Generation [To define and generate screens that support retrieval and presentation, update of data]

Engineering Support: [For analyses, design, modeling, development, simulation for users and environments. Includes CAD services, decision support tools, expert-system shells]

Computer-Aided Design [High-precision drawing and modeling tools]

Decision Support [Interactive modeling and simulation tools for analyzing alternative decisions]

Expert System [Artificial intelligence capabilities based on knowledge or inference engines]

Modeling and Simulation [Capabilities to capture or set object characteristics]

Environment Management: [Integrate and manage the execution of platform services]

Batch Processing [Capability to queue work, sequence managing, asynchronous tasks]

Computer-Based Training [Provide integrated training environment, online documentation, help files, context sensitive definitions]

Information Presentation and Distribution [To manage distribution and presentation of information, can store, archive, prioritize, restrict, recreate information]

Transaction Processing [On line capture and processing of information in an interactive exchange with the user]

Multimedia: [Manipulate and manage information consisting of text, graphics, images, video, audio].

Audio Processing [Capture, compose, and edit audio information]

Document Processing [Create, edit, merge, format documents, scanning]

Electronic Publishing [Photographic images, color graphics, advanced formatting and style features]

Image Processing [Capture, scan, create, and edit images]

Map Graphics [Map manipulating, creating entity symbology, create, edit, compose drawings, symbols, maps]

Multimedia Processing [compress, store, retrieve, modify, sort, search, print hypermedia, optical storage technology, data compression, digital storage techniques]

Video Processing	[Capture, compose, edit still graphics, title generation,
Text Processing	[Create, edit, merge, and format text]

D.2.2 Application Platform Services

Software Engineering Services: [For development and maintenance tools of software applications]

Bindings	[Access to applications-bindings and object code linking]
Computer-Aided Software Engineering (CASE)-Tools Environment	[Requirements specifications and analysis, design, testing, and prototyping, configuration control]
Language	[Syntax, semantic definitions, shell/script/procedural/ object-oriented/3rd-generation languages]
Software Life-Cycle Processes	[Activities, methods, processes, transformations, development/maintenance, all phases including post-deployment support]

User Interface Services: [How users interact with an application]

Character-Based User Interface	[Command line, menu-driven, keyboard input, no graphics]
Graphical Client-Server	[Relationships between client and server processes, graphical user interface display processes]
Object Definition and Management	[Characteristics of display elements-color, shape, size, movement, graphics context]
User Interface	[Interaction with applications, how to gain access to applications programs/OS/utilities, i.e., menus, screen designs, keyboard Commands, command language, the way user interacts with computer]
Window Management	[How windows are created, moved, stored, retrieved, removed, and related to each other]

Data Management Services: [Management of data independent of processes that create or use it]

Data Dictionary/Directory	[Define and obtain data in DB, access and modify data, internal/external formats, integrity, security rules, standardization and registration of data elements, data sharing and interoperability]
Database Management System	[Data management in a distributed system, data administration, controlled access to, and modification of,

structured data, create, populate, move, backup, restore, and archive DBs]

Transaction Processing [Support for online capture and processing of information in interactive exchange with the user]

Data Interchange Services: [Support for interchange of information between applications and external environment]

Characters and Symbols [Interchange of character sets/fonts/date/time representation]

Compression [Algorithms for data (text, still images and motion mages) storage and exchange]

DoD Applications [Functional areas unique to DoD mission that are not standardized]

Document Interchange [Specifications for encoding data (text, pictures, numerics, special characters), and logical/physical structure of documents]

Hardware Applications [For data interchange between non-homogeneous hardware, bar-coding, optical disk-handling, graphics device interface]

Mapping [Formats and facilities for machine-readable graphics-based mapping, charting and geospatial data]

Optical Digital Technologies [For light and optical technologies to capture, encode/ decode, store data]

Product Data Interchange [Specifications that describe drawings, documentation and data for product design and manufacturing, for geometric and non-geometric data]

Raster/Image Data Interchange [Handling/manipulating raster graphics and images, pixel-by-pixel representation or imagery data exchange and attachments to images]

Technical Data Interchange [Standards for interchange of graphics data, vector graphics, technical specifications]

Graphics Services: [Standards for creating and manipulating pictures]

Device Interfaces [Services for accessing graphics devices (monitors/ printers)]

Raster Graphics [Image representation on a matrix of dots, monochrome/ gray-scale/color bit maps, creation by scanner, cameras, and color paint software packages]

Vector Graphics [Graphical objects as sets of end points for lines, curves, and geometric shapes, geometric knowledge and display lists, shape integrity]

Network Services: [To support distributed applications requiring data access and applications interoperability]

Application-Oriented	[Functions and interfaces on network and communications system protocol software used by applications]
Subnetwork Technologies	[LANS and other data communications services concerned with physical and data-link layers (1and2 OSI Model)]
Transport-Oriented	[End-to-end transmission of data across network and end to end reliability, end-end error detection/recovery, flow control and monitoring quality of service]

Operating System Services: [Core services to operate application platform and provide an interface between application software and platform]

Clock/Calendar	[Mechanisms for measuring and maintaining all time]
Fault Management	[Prevention, isolation, diagnosis, correction whenever abnormalities occur]
Kernel Operations	[Low-level services to create/manage processes, execute programs, manage files/directories, and control I/O processing to and from peripheral devices]
Media Handling	[Disk and tape formatting of data and interchange of data]
Operating System Object	[Rules for creating, deleting, managing objects]
Real-Time Extension	[Support for event-driven processes, interrupt processing]
Shell and Utilities	[Operator-level services-comparing, printing, displaying file contents, file management, sorting data, displaying file contents]

Internationalization Services: [Services and interfaces to allow users to define, select, change between culturally related application environments, different market segments]

Character Sets and Data Representation	[Input, store, manipulate, retrieve, communicate, data independent of coding scheme, GUI modifications, syntax-consistent-semantic-independent]
Cultural Convention	[Support for local rules and conventions]
Native Language Support	[Support for more than one language for local character sets]

Security Services: [For protection and separation of sensitive information]

Access Control	[Unauthorized use of information-system resources]
Architectures and Applications	[Security architecture and placement of security into specific applications]
Authentication	[Unique and proper identification and authentication of system elements]

Availability	[Assurance of timely and regular communications, graceful degradation]
Confidentiality	[Ensures that data is not made available to unauthorized individuals or computer processes]
Integrity	[Protection of system through open-system integrity, network integrity and data integrity, data is not altered or destroyed in an unauthorized manner]
Non-Repudiation	[Non-denial of origin or delivery of data, validation of source software packages and hardware]
Security Labeling	[Accuracy and integrity of security labeling]
System Management	[Certification, accreditation, and risk management, alarm reporting, audits, cryptographic key management]

System Management Services: [State management, configuration control, performance, monitoring, fault monitoring, user/group management, usage monitoring]

Configuration Control	[Identification, control, status accounting, verification, software distribution, license management]
Fault Monitoring	[Event management and network recovery, loss or incorrect operation of system components]
Information System Security Management	[Installation, maintenance, enforcement of information domain and system security policy rules]
Other Management	[Database management and administration, print management]
Performance Monitoring	[Performance aspects of hardware, software, network components, system resource management, device management]
State Management	[Monitoring, maintaining, and changing state of system]
Usage Monitoring and Cost Allocation	[Management of licensing, system cost management, system resource allocation]
User/Group Management	[Interfaces for administering users and groups, implementation of management policies across a system, group/user access to applications]

Distributed-Computing Services: [Support for applications distributed or dispersed among systems in a network yet maintain cooperative processing environment]

Client-Server	[Computing services partitioned into requesting processes (clients) and providing processes (servers) on same/ distributed platforms]
Object	[Definition, instantiation, interaction of objects in a distributed environment, OS bindings, message transport and delivery, data persistence]

Remote Access	[Location transparency functionality, access to appropriate systems resources (files, data, processes)]

D.2.3 External Environment

External Environment: [Entities with which Application Platform exchanges information]

Human Users	[People]
Information Interchange	[removable memory]
Communications	[Telephones, networks, cabling, packet-switching equipment]
Doctrinal Mechanisms	[Physical, administrative, personnel; provide for security protection of information-system components in external environment]

APPENDIX E - CASE STUDY/EXAMPLES (DOMAIN EXAMPLES)

A number of examples are being evaluated for inclusion into this appendix. Their intent is to illustrate actual usage of the DoD TRM in assisting to provide interoperability solutions.

The following examples are being considered for inclusion in this appendix:

[Army TACOM Real-Time System Example]

[Army – Networks Protocol Example]

[Department of Commerce Patent Trademark Office Development Example]

[Air Force and Aerospace Technical Architecture/Standards Profile Example]

TCP/IP TO TRM CASE STUDY

Objective of Case Study

This case study will show how the Transmission Control Protocol/Internet Protocol (TCP/IP) communication suite of protocols uses a layering implementation that can be modeled via the Interface Side of the Department of Defense (DOD) Technical Reference Model (TRM), which uses a set of defined direct and logical interfaces. This case study provides a very general description of how TCP/IP relates to the DOD TRM, and it does not provide an in-depth explanation of TCP/IP (see references below for further information about TCP/IP).

Rationale/Purpose

The DOD TRM is an evolution between the TAFIM and the General Open Architecture (GOA) model. The TAFIM defines Services within the layers of the POSIX model, and the General Open Architecture (GOA) standard, SAE AS4893, defines the Interfaces between the layers. These layers share information with each other through a defined set of interfaces; protocols in the case of TCPIP. The DOD TRM stack (stacked layers) begins with the physical hardware layer and proceeds through a set of layers to the application layer via a precise set of direct and logical interface definitions that clearly identify the separation of boundaries between the layers. A layered structure provides the following attributes: layer portability, interoperability between associated layers, plug-and-play implementations, and affordability via contractor-competitive competition for the development of each of the independent layers.

The purpose of this case study is to show that the TCP/IP suite of communication protocols can be modeled using the DOD TRM. Note: An example of military system that uses TCP/IP is the Combat Net Radio (CNR).

Description/Process Used

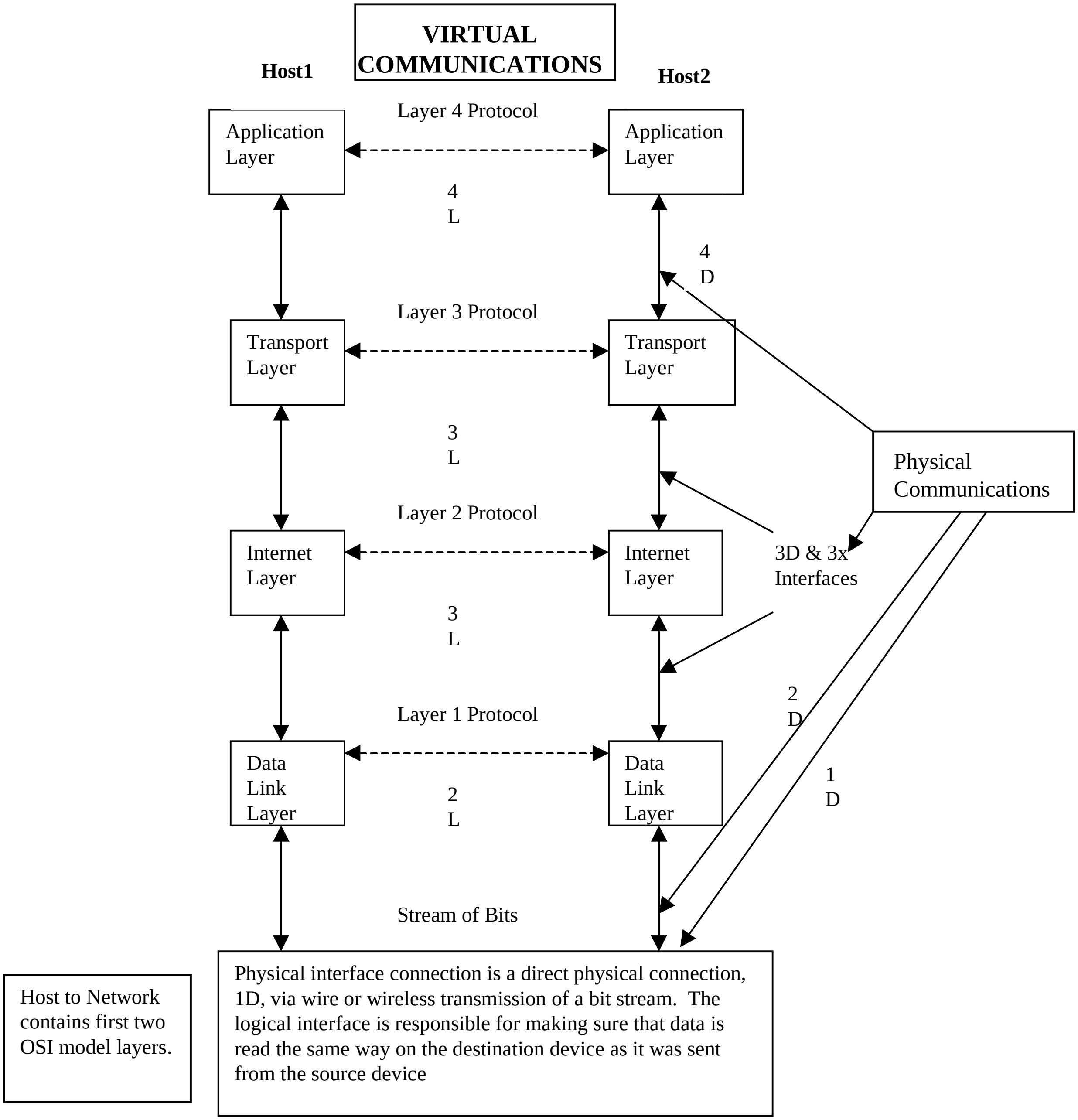
In the DOD TRM to TCP/IP relationship diagrams below, the models display a layered structure. The DOD TRM layers define boundaries that should be used when developing computer hardware and software interfaces. The DOD TRM abstract model helps a software developer recognize interfaces and modularize the software into layers. These layers then pass information among other components and systems.

To develop a communication suite of protocols that are portable and easily maintained, the DOD TRM is the modeling tool that can help identify the hardware/software layers needed in the passing of information. A concern in passing information is how to identify and define the software data parameters needed to transfer the information from one platform to another platform successfully keeping in mind the separation of boundaries to the input layer, output for that layer, and the logical interface understanding between adjacent layers or systems. A software developer must also look at how the passed information will be protected and verified as correct upon receipt. Header data, bit insertion techniques, and various algorithms are used to ensure that application data within a bit stream is not corrupted. Also within a communications protocol, the receiving layers should send acknowledgements back to the sender of the information letting that layer know the passed information was successful. If not successfully passed, the sending layer will retransmit the information.

An example of such a communications protocol is the Transmission Control Protocol (TCP) Internet Protocol (IP). TCP/IP uses four stacks or layers to transfer data, because the physical layer and data link-layer are combined (see figure 3). In the figure 1, these two layers have been broken out to show the distinction between the layers.

The purpose of a layer is to offer certain services to the higher layers, shielding those layers from the details of how the services are actually implemented. The Application Layer on one-machine carries on a conversation with the Application Layer Y on another machine via the protocol definition (logical interface definition). A resulting communications protocol stack is formed when data is passed from each individual layer from the lowest layers to the highest layers and the highest layers to the lowest layers. Figure 1 below shows a protocol stack formation, and the virtual commutations between host1 to host2.

The virtual communications between host1 and host2 is the logical interface classes, which establishes the communications understanding or protocol understanding between the layers. The physical communications is the direct-software-layer-algorithm-to-direct-software-layer-algorithm interface, which passes and receives the defined software data parameters based on the logical interface definition. In the TCP/IP communications stack, the software layer algorithms are created based on the protocol definition, which determines the data stream format for the input and output of each layer. Each software-layer algorithm implementation (what is actually happening inside a layer) is independent from other software layer algorithms. Virtual communications (logical interface) would not exist without direct interfaces, and the logical interface definition determines the data parameters that the direct interface algorithms utilize.



Host to Network contains first two OSI model layers.

Physical interface connection is a direct physical connection, 1D, via wire or wireless transmission of a bit stream. The logical interface is responsible for making sure that data is read the same way on the destination device as it was sent from the source device

TRM model interfaces applied to TCP/IP Communications Stack

Below is a description of how the TCP/IP packets, data stream, of information are separated into columns of information, which form the layers in the TCP/IP stack.

In the TCP/IP communication protocol, the architecture is based on the Internet datagram definition. The makeup of the datagram, data stream, is a stream of bits (1’s and 0’s). This stream of bits is further defined by bytes. There are 8 data bits to one byte. The Internet datagram is defined as:

Physical/Data Link	IP Layer Data	TCP or UDP Layer	Application Data	Frame Check Sequence

Note: The Frame Check Sequence is used as error checking in data transmission, and UDP is used for error-checking when sending isolated messages to another system.

A datagram is a finite-length packet with sufficient information to be independently routed from source to destination without reliance on previous transmissions. Each of the separately defined byte lengths represents data that the software layer algorithm understands via the protocol definition.

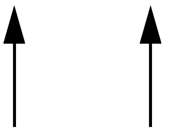
Physical Layer

The DOD TRM physical layer is responsible for generating and actually passing the physical electrons, stream of information, from sending hardware component to the receiving hardware component. The physical layer contains the hardware, sensors, microcontroller, microprocessors, data buses, electromagnetic waves, and electrical interface requirements, which is the 1D direct interface class definition.

Logical Interface

An example of the logical interfaces associated with the physical layer bit stream of information being passed from the sending hardware to the receiving hardware is shown below.

011111 11011111 010 Data Bits
0111110110111110010 Stream of Information being Transferred



Extra Bits Inserted and Removed by the Hardware

The sending hardware inserts a zero bit (0) after every five consecutive one-bit pattern (1). This procedure is called zero-bit insertion or bit stuffing. The receiving hardware will remove the inserted zero bits upon receipt of the data stream. The two hardware components must have already defined the logical interface definition, zero-bit insertion method, before transferring the data steam. So, the sending and receiving hardware will have an understanding on how they will communicate with each other before sending and receiving information data. This logical interface definition is normally used so that duplication for the user data (application data) is not corrupted from the sending hardware to the receiving hardware.

Direct Interface

The direct interface is a bus connecting the two hardware pieces, or the electromagnetic wave form that is transferred through the atmosphere connecting the two hardware pieces.

Data-Link Layer

The task of the data-link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Various framing methods are used, including character count, character stuffing, and bit stuffing. Data-link protocols can provide error-control to retransmit damaged or lost frames.

Internet Protocol Layer

The Internet Protocol (IP) layer routes data between the hosts or other systems (Figure 1). The information data may be passed to a single network, or it may be relayed across several networks on the Internet. IP address data routes its traffic without caring which application-to-application interaction a particular datagram belongs to.

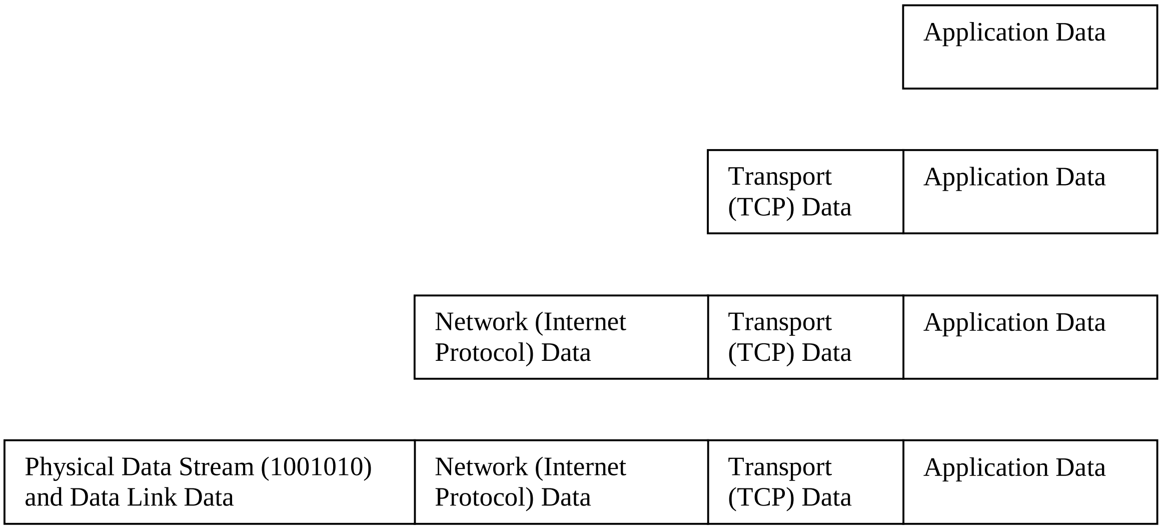
Transmission Control Protocol (TCP)

The TCP provides a reliable data connection service to applications. TCP has an algorithm, which guarantees that data is error-free, complete, and in proper sequence for recombining packets of data. TCP ensures that data is sent and received by sending an acknowledgement message between the sending TCP layer system and receiving TCP layer system. It is a logical interface or virtual communications interface.

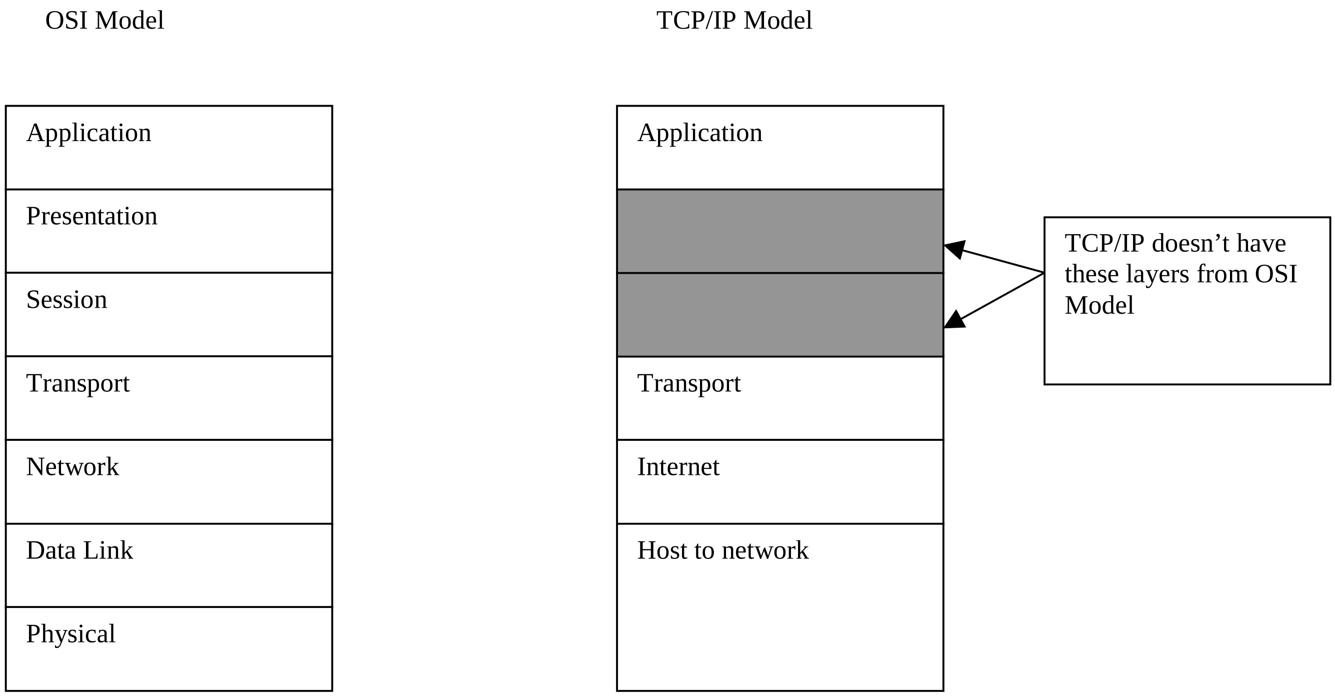
Application Layer

The Application Layer Services used by TCP/IP are: File Transfer Protocol, Simple Mail Transfer Protocol, Telnet terminal access, Domain Name System (DSN) directory services, and program-to-program communications.

The two diagrams below (Figures 2 and 3) show how the information data can be separated into segments and represented as a layers/stacks.



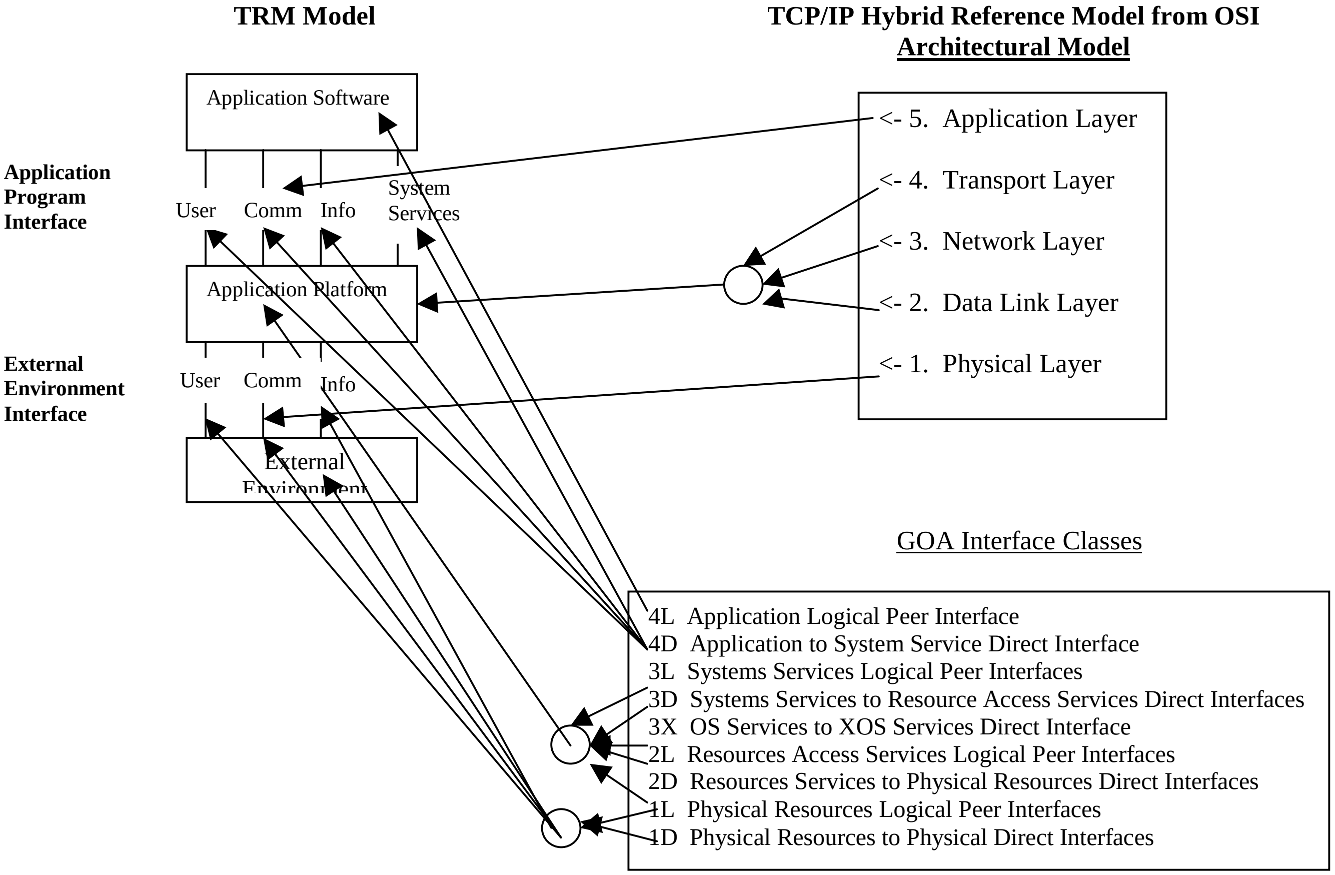
RELATIONSHIP



Results/Findings

In the comparison between the DOD TRM, Open Systems Interconnect (OSI), GOA, and TCP/IP, the TCP/IP model has been broken down using the DOD TRM and OSI model concepts. The TCP/IP model does not make a distinction between the physical and data link layers. Each layer in the TCP/IP model uses the fields in the datagram above to check for error transmissions and router destinations while protecting the application data. Each of the defined fields, bytes, is stripped away via a particular layer algorithm as defined by the protocol being used. When the remaining datagram, data stream, reaches the application layer, the bit stream contains some header information for error-checking and the application data. The application software can use the application data without having any knowledge of how the data arrived.

The figure 4 below is another view of the relationship among the basic DOD TRM model, TCP/IP, and the DOD TRM interface classes. The TCP/IP model can easily be represented via DOD TRM model. The DOD TRM thus helps in determining design and software implementations in developments. The DOD TRM also establishes a common language among designers, when they discuss their particular software layer implementations.


TRM Model
TCP/IP Hybrid Reference Model from OSI
Architectural Model
Application Software
Application
Program
Interface
User
Comm
Info
System
Services
Application Platform
External
Environment
Interface
User
Comm
Info
External
Environment
<- 5. Application Layer
<- 4. Transport Layer
<- 3. Network Layer
<- 2. Data Link Layer
<- 1. Physical Layer
GOA Interface Classes
4L Application Logical Peer Interface
4D Application to System Service Direct Interface
3L Systems Services Logical Peer Interfaces
3D Systems Services to Resource Access Services Direct Interfaces
3X OS Services to XOS Services Direct Interface
2L Resources Access Services Logical Peer Interfaces
2D Resources Services to Physical Resources Direct Interfaces
1L Physical Resources Logical Peer Interfaces
1D Physical Resources to Physical Direct Interfaces

Expectations

The TCP/IP communications protocol is an example of how to define, using the DOD TRM, interfaces within an system. Defining the interfaces between layers makes the software portable. A common understanding (datagram definition) should be established between system layers for the purpose of interoperability. The DOD TRM provides the layer concept needed to support interoperability, portability, and affordability. The DOD TRM is a good tool to use for computer and software developers.

Many domains (commercial, government, private citizens, countries) use the TCP/IP protocol in their communications with each other. Interoperability between these domains is achieved because they use the same standard (TCP/IP standard). The Joint Technical Architecture (JTA) accomplishes the same interoperability goals, mandating standards to be used by the weapons community.

Definitions

Destination Address – This field contains the destination information (used by the network Layer).

Source Address – This field contains source information (used by the network Layer).

User Data field – This field contains data used by the application (used by application).

Frame Check Sequence – Error-checking for data transmission

This Service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented.

Protocol is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer. Layers use protocols in order to implement their service definitions.

Host to Network (first two layers):

1. The physical layer is concerned with the transmission characters of wire, fiber-optics, and wireless communications. It passes the stream of bits in the form of a datagram.

2. The data link layer delimits the start and end of frames to be used by the network layer.

The network layer is concerned with getting packets from the source to the destination.

The transport layer provides reliable, cost-effective data transport from the source machine to the destination machine, independent of the physical network.

The application layer is concerned with security, naming within the Internet, network management, and the applications such as electronic mail, net news, multimedia, and Web.

References

- TCP/IP (Architecture, Protocols and Implementation) by Sidnie Feit
- Computer Networks, Third Edition, by Andrew S. Tanenbaum
- DoD TRM Version 1.0
- SAE AS4893

Case Study on Mapping of Defense Information Infrastructure (DII) Common Operating Environment (COE) Segments to DoD Technical Reference Model (TRM) Services

Purpose and Objective of the Case Study

The Defense Information Infrastructure (DII) Common Operating Environment (COE) is an infrastructure for building interoperable systems across mission-area applications using a set of guidelines, standards and specifications implemented through a collection of reusable components or segments. In the DII COE Integration and Runtime Specification (I&RTS), a segment is defined as a collection of one or more software and/or data units most conveniently managed as a unit of functionality. Building a target system includes combining COE components with mission-specific software.

The mapping document categorizes the DII COE segments into the service areas defined in the DoD TRM. Each COE segment is mapped into a DOD TRM service based on the segment's functionality as determined from DII COE and COTS documents. Although the DOD TRM provides for both a services and an interfaces view, at this time the mapping contains only the services view.

Rationale

In the DII COE documentation, a conceptual correspondence already existed between the DOD TRM and the DII COE, illustrated in figures showing the DII COE services and architecture. As a logical follow-on to the notional mapping already in the DII COE documentation, a finer granularity of mapping was developed of the DII COE segments to the services defined in the DOD TRM. The mapping provides a common foundation for viewing diverse components from a conceptual perspective, and helps to understand the roles of the various COE segments. It can be used not only to contrast DII COE with other system architectures but also to provide a basis for correlating interoperablility areas between the DII COE and other system architectures.

Description

The COE segments referenced in the mapping were derived from the DII COE 3.4 Build Lists. The purpose of each segment in the build list was identified, and based on the segment’s purpose, it was mapped to a DOD TRM service. The software version description document and the DII COE Integration and Runtime Specification (I&RTS) document were used to obtain a segment’s function. If a software version description document was unavailable for a segment and the I&RTS provided no descriptive information, the World Wide Web was searched for information related to the segment.

The mapping document contains a listing of the DOD TRM service areas, each followed by the descriptions from Section 4.4 of the DOD TRM. Below each DOD TRM service are the COE segments mapped to that service. With each segment is a description of the segment, with the source for the description identified. The host platform(s) on which each segment is available are also listed.

Results/Findings

Most of the DII COE segments were easily mapped to a single DOD TRM service area, but some of the segments could be categorized into more than one service area. To minimize ambiguity, the primary purpose of the segment was identified and used to categorize it into a single DOD TRM service area.

For example, the DocViewer segment allows users to view documents installed locally or documents loaded on any document server. This segment was mapped to DoD TRM Section 4.4.1.2.1, Multimedia Services.

Similarly, the Application Framework segment is a client to the Joint Mapping Toolkit – Visualization segment that provides a framework for application segments that share a common tactical display. The DII Motif Style Segment tailors and extends basic Motif features in accordance with Version 3.0 of the DII COE User Interface Specification. Both of these segments were mapped to DoD TRM 4.4.2.1.2 User Interface Services.

As another example, the Object Request Broker is an intermediary that coordinates and manages the requests between clients and servers. This segment was mapped to DoD TRM 4.4.2.4.4 Distributed Computing Services.

There were some segments that did not appear to map directly to an existing service area. For each of those segments, a decision was made to map the segment into the DOD TRM service area that seemed most closely related to the purpose of the segment. The segments where this tailoring of the model was required included the Software Development Kits (SDK), the Alerts, and the data segments.

The SDKs provide examples and libraries that a developer can utilize to develop or expand the function of a COE segment. Each SDK was categorized into one of the DOD TRM service areas based on the functionality developed by utilizing the SDKs. For instance, the Universal Communications Processor (UCP) SDK provides the development libraries, including scripts, data, and sample sources, for third party UCP developers to build client applications for the UCP engines. Since the UCP mapped to the DoD TRM Section 4.4.2.1.6, Communications Services, the UCP SDK was also mapped to that service area.

The Alerts segment provides a generic mechanism for the sending and receiving of alert messages between processes. It provides applications with the ability to register specific events which will generate visual/audio/log alerts to the operator during the operation of the system. This segment was mapped to the DoD TRM Section 4.4.2.1, System Services, since the description of the services in that service area was closest to the functionality of the alerts.

The data segments can be classified into account groups, templates, and data used by runtime software. They don't provide any functionality by themselves, but rather provide a mechanism for creating or modifying the runtime environment. The account groups, which provide samples for accessing and customizing the user interface, were mapped to the User Interface Services. The templates provide examples of how to customize the behavior of a parent segment and were mapped to the same DOD TRM service area as the parent segment. The data segments were mapped to the same DOD TRM service area as the segment that utilized the data at runtime. For example, the MIL-STD-2525 Symbology data segment, which contains Computer Graphics Metafiles (CGM) and a menu hierarchy that allow the construction of MIL-STD-2525A icons, was mapped to DoD TRM Section 4.4.1.2.1, Multimedia Services.

Conclusions

The DOD TRM is very useful in gaining a better understanding of the roles and functions of the DII COE segments. Moreover, the mapping process also provides useful feedback on the DOD TRM itself. The mapping provided valuable insight into the tailoring process, and may even provide input in possible future updates of the DOD TRM.

References

Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS), Version 3.1, Defense Information Systems Agency, 1 October 1998

Department of Defense Technical Reference Model (DoD TRM) Version 1.1 Coordination Draft, Defense Information Systems Agency, 15 July 1999

Mapping of Defense Information Infrastructure (DII) Common Operating Environment (COE) Segments to DoD Technical Reference Model (TRM) Services, Aerospace Report No. ATR-99(3583)-1, The Aerospace Corporation, 26 August 1999

APPENDIX F - CHANGE PROPOSALS AND CONFIGURATION MANAGEMENT

F.1 INTRODUCTION

Changes to the DoD TRM will occur through changes to the document. This appendix provides guidance for submitting of proposed changes. These proposed changes should be described as specific wording for line-in/line-out changes to a specific part of the document.

Use of a standard format for submitting a change proposal will expedite processing changes. The format for submitting change proposals is shown in Section F.2. Guidance on using the format is provided in Section F.3.

A Configuration Management contractor is managing the receipt and processing of DoD TRM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown in Section F.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **trmwg@ncr.disa.mil**

Mail: **DoD TRM**

ARTEL
1893 Preston White Drive Reston, Virginia 20191 Fax (703) 620 4262

F.2 DOD TRM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

F.3 FORMAT GUIDANCE

The format in Section F.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific Section, multiple-change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TRMWG project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**F.2 a)** of the format identifies that individual. The information in the **Document Identification** part of the format (**F.2 b**) is self-evident, except that volume number will not apply to the CMP or PMP. The proposed changes will be described in the **Proposed Change #** parts (**F.2 c, F.2 d, or F.2 n**) of the format.

In the Proposed Change # parts of the format, the section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) further identifies where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field, the submitter identifies the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, is helpful. An example of input for this field is: “Delete the last sentence of the second paragraph of the section and replace it with the following sentence: ‘The working baseline will only be available to the TRMWG project staff.’”

The goal is for the commentator to provide proposed wording appropriate for insertion into a document without editing (i.e., a line-out/line-in change). The F.2 c (5), F.2 d (5), or F.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as “industry practice” carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific DoD TRM document may be provided in F.2 c (6), F.2 d (6), or F.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity, these comments may not result in change to the document.